

Maria do Carmo Proença Caseiro Brás

**PROBLEMAS DE PROGRAMAÇÃO MATEMÁTICA
COM RESTRIÇÕES LINEARES DE EQUILÍBRIO
E APLICAÇÕES**

Lisboa
2006

**PROBLEMAS DE PROGRAMAÇÃO MATEMÁTICA
COM RESTRIÇÕES LINEARES DE EQUILÍBRIO
E APLICAÇÕES**

Dissertação apresentada para obtenção do grau de
Doutor em Matemática na especialidade de Inves-
tigação Operacional, pela Universidade Nova de Lisboa,
Faculdade de Ciências e Tecnologia.

Agradecimentos

Um grande número de pessoas contribuiu directa ou indirectamente para a realização desta tese. Demonstrar a minha gratidão por todos quantos colaboraram para a sua elaboração é tarefa que muito me apraz.

Ao Professor Doutor Joaquim João Júdice cujo estímulo, apoio, orientação e entusiasmo contagiante foram absolutamente essenciais. Para além de um supervisor incansável tornou-se um grande amigo, sempre com uma palavra encorajadora mesmo nos momentos menos bons desta longa caminhada.

À Professora Doutora Isabel Ribeiro pelo "software" cedido, por toda a sua preciosa ajuda, pela disponibilidade constante e pela amizade que se foi construindo com o desenvolvimento desta tese.

Aos Professores Doutores Reiner Horst e Luís Merca Fernandes pelas suas importantes colaborações na elaboração desta tese.

Ao Departamento de Matemática da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa e em particular ao Professor Doutor Ruy Costa por todo o apoio prestado no meu trabalho de doutoramento.

À minha família que foi a motivação e exemplo de dedicação, sem a qual me seria certamente impossível a realização deste trabalho. Destaco o meu filho Duarte, cujo nascimento me proporcionou inextinguível felicidade, sendo fonte diária de energia para a conclusão deste estudo.

Aos colegas da FCT/UNL pelo companheirismo e sincero interesse que sempre demonstraram pelo andamento do meu trabalho.

A todos os que de uma forma ou de outra contribuíram para o progresso desta tarefa, o meu Muito Obrigada!

Resumo

Neste trabalho é desenvolvido um algoritmo enumerativo paramétrico de otimização global para a resolução de Problemas de Programação Matemática com Restrições de Equilíbrio ou de Complementaridade (*MPEC*). A comparação com outras técnicas globais da literatura é efectuada para um leque variado de problemas, de modo a poder avaliar a eficiência do processo proposto. A utilização de algoritmos de *MPEC* para a resolução de alguns problemas de otimização global é o outro grande objectivo desta tese. Nesse sentido são introduzidas novas formulações de programas bilineares e lineares complementares como *MPECs*. São ainda analisadas e discutidas formulações *MPEC* para o problema de programação linear inteira 0-1, para a determinação do Conjunto Independente Máximo de um Grafo (*MIS*) e para a estimação do Número de Condição de uma Matriz. Para o problema *MIS* é desenvolvido um algoritmo de ramificação e limitação, baseado na decomposição de uma função quadrática numa diferença de duas funções convexas (DC). Finalmente é introduzida uma técnica *MPEC* local para a estimação do número de condição com a norma l_1 e é estabelecido para matrizes de *Minkowski* que o número de condição nessa norma pode ser estimado com apenas um sistema de equações lineares. Em todos os desenvolvimentos houve uma grande preocupação em testar as novas formulações e algoritmos com problemas conhecidos da literatura, de modo a aferir da qualidade e interesse dessas propostas.

PALAVRAS CHAVE: Otimização Global, Complementaridade, Programação Inteira, Conjunto Independente Máximo, Número de Condição

Abstract

In this work a parametric enumerative algorithm is developed for finding a global minimum of a Mathematical Programming Problem with Equilibrium or Complementarity Constraints (*MPEC*). A comparison with other global techniques from the literature is done for different problems, in order to evaluate the efficiency of the proposed procedure. The use of *MPECs* algorithms to solve some global optimization problems is another main objective of this thesis. To do this, new formulations of Bilinear and Linear Complementarity problems as *MPECs* are introduced. It is also analyzed and discussed *MPEC* formulations for the Integer Linear 0-1 programming problem, the Maximal Independent Set of a graph (*MIS*) and the Matrix Condition Number Estimation. A Branch and Bound algorithm, is developed for processing the *MIS* problem, which is based on the decomposition of a quadratic function into a difference of two convex functions (DC). Finally it is introduced a local *MPEC* technique to estimate the condition number in the l_1 norm and it is established that this condition number can be estimated with only a linear system of equations if the matrix is Minkowski. The new formulations and algorithms have been tested with different problems from the literature, in order to highlight the quality and interest of these proposals.

KEYWORDS: Global Optimization, Complementarity, Integer Programming, Maximal Independent Set, Condition Number

Résumé

Dans ce travail, nous développons un algorithme énumératif paramétrique d'optimisation globale pour la résolution de Problèmes de Programmation Mathématique avec Restrictions d'Équilibre ou de Complémentarité (*MPEC*). Nous effectuons, pour un large éventail de problèmes, la comparaison avec d'autres techniques globales existantes dans la littérature, afin d'évaluer l'efficacité du processus proposé. L'autre grand objectif de cette thèse est l'utilisation d'algorithmes de *MPEC* pour la résolution de quelques problèmes d'optimisation globale. Pour cela, nous introduisons nouvelles formulations de programmes bilinéaires et linéaires complémentaires comme *MPECs*. Nous discutons et analysons aussi des formulations *MPEC* pour le problème de Programmation Linéaire Entière 0-1, pour la détermination de l'Ensemble Indépendant Maximal d'un Graphe (MIS) et pour l'estimation du Conditionnement d'une Matrice. Pour le problème MIS, nous développons un algorithme de ramification et limitation, basé sur la décomposition d'une fonction quadratique comme la différence de deux fonctions convexes (DC). Finalement, nous introduisons une technique *MPEC* locale pour l'estimation du conditionnement en norme l_1 et nous établissons que, pour les matrices de Minkowski, le conditionnement dans cette norme peut être estimé simplement à l'aide d'un système d'équations linéaires. Nous nous sommes attachés à développer ces notions en testant les nouvelles formulations et algorithmes sur des problèmes connus de la littérature, afin de mesurer leurs qualités et leurs intérêts.

MOTS-CLÉS : Optimisation Globale, Complémentarité, Programmation Entière, Ensemble Indépendant Maximal, Conditionnement.

Notações

\mathbb{R} : Recta real.

\mathbb{R}^n : Espaço vectorial euclidiano real de dimensão n .

\mathbb{R}_+^n : Octante não negativo do espaço vectorial euclidiano \mathbb{R}^n .

$\mathbb{R}^{m \times n}$: Espaço das matrizes reais de ordem $m \times n$.

\mathbb{N} : Conjunto dos números naturais.

$[a, b]$: Intervalo fechado.

$|A|$: Número de elementos de um conjunto finito A .

\emptyset : Conjunto vazio.

$|t|$: Valor absoluto do número real t .

λ^k : Número real λ na iteração k .

x : Vector coluna.

x_i : Componente i do vector x .

$sign(x)$: Vector de componentes 1 e -1 associadas ao sinal das componentes x_i .

e : Vector de componentes unitárias.

x^k : Vector x na iteração k .

$\|x\|_1$: Norma l_1 do vector x .

$\nabla f(x)$: Vector gradiente de f em x .

I : Matriz identidade.

e^j : j -ésima coluna da matriz identidade.

A^T : Matriz transposta de A .

- A^{-1} : Matriz inversa de A .
 a_{ij} : Elemento da linha i e coluna j da matriz A .
 $A_{i.}$: Linha i da matriz A .
 $A_{.j}$: Coluna j da matriz A .
 $\|A\|_1$: Norma l_1 da matriz A .
 $\|A\|_\infty$: Norma infinita da matriz A .
 $\text{diag}(a)$: Matriz diagonal cujos elementos diagonais são as componentes do vector a .
 $c(A)$: Característica da matriz A .
 $\rho(A)$: Raio Espectral da matriz A .
 $\text{cond}(A)$: Número de condição da matriz A .
 LCP : Problema Linear Complementar.
 $ALCP$: Problema Linear Complementar Aumentado.
 $GLCP$: Problema Linear Complementar Generalizado.
 NLP : Programa Não Linear.
 LP : Programa Linear.
 QP : Programa Quadrático.
 BPP : Programa de Dois Níveis.
 $MPEC$: Programa Matemático com Restrições de Equilíbrio (ou Complementaridade).
 MIS : Problema do Conjunto Independente Máximo de um grafo.
 $G(V, E)$: Grafo G com V o conjunto de vértices e E o conjunto de arestas.
 \bar{G} : Grafo Complementar de G .
 A_G : Matriz de Adjacência do Grafo G .
 $\alpha(G)$: Número de Independência do Grafo G .
 $\omega(G)$: Número de Clique do Grafo G .
 $\theta(G)$: Número de Cobertura por Cliques do Grafo G .
 $\gamma(G)$: Número Cromático do Grafo G .

Classes de Matrizes

PSD : Semi-Definida Positiva.

NSD : Semi-Definida Negativa.

IND : Indefinida.

PD : Definida Positiva.

$RSuf$: Suficiente por Linhas.

$CSuf$: Suficiente por Colunas.

Suf : Suficiente.

C : Copositiva.

SC : Copositiva Estrita.

P : Matriz com Menores Principais Positivos.

P_0 : Matriz com Menores Principais não Negativos.

Índice

Introdução	1
1 Programação Matemática com Restrições de Equilíbrio	5
1.1 Problema Linear Complementar	5
1.2 Problema de Programação Matemática com Restrições de Equilíbrio (<i>MPEC</i>)	8
1.3 Formulação <i>MPEC</i> do Problema de Dois Níveis	11
1.4 Formulação <i>MPEC</i> de Programação Bilinear	13
1.5 Formulações <i>MPEC</i> do Problema Linear Complementar	14
2 Algoritmos para Programação Matemática com Restrições Lineares de Equilíbrio	21
2.1 Algoritmo Enumerativo para o <i>GLCP</i>	21
2.2 Algoritmo Complementar de Restrições Activas	24
2.3 Algoritmo Sequencial Complementar	29
2.4 Algoritmo Complementar de Pesquisa em Árvore	31
2.5 Algoritmo Enumerativo Paramétrico	33
2.6 Experiência Computacional	36
3 Programas Lineares Inteiros	45
3.1 Formulação <i>MPEC</i> de Programas Inteiros	45
3.2 Algoritmos <i>MPEC</i> para Programação Inteira	47
3.3 Experiência Computacional	52

4	Número de Independência de um Grafo	61
4.1	Definição do Problema	61
4.2	Formulações Quadráticas	63
4.3	Formulações <i>MPEC</i>	65
4.4	Algoritmo de Diferenças Convexas	69
4.5	Determinação de um Conjunto Independente	78
4.6	Algoritmo Sequencial Complementar	80
4.7	Experiência Computacional	81
5	Estimação do número de condição	87
5.1	Definição do Problema	87
5.2	Estimadores de <i>Hager</i> e de <i>Higham</i>	88
5.3	Formulação <i>MPEC</i>	91
5.4	Experiência Computacional	96
	Conclusão	103
	Bibliografia	105

Introdução

A optimização global constitui um dos assuntos de grande interesse da comunidade científica, devido às suas inúmeras aplicações em várias áreas da ciência, economia e engenharia. As dificuldades na resolução de problemas de optimização global prendem-se com o facto das técnicas usuais de optimização não linear determinarem apenas óptimos locais. Como não existe em geral um critério para decidir quando é que um óptimo local é global, os métodos convencionais de optimização não linear não são capazes de localizar e identificar óptimos globais.

Um programa matemático com restrições de complementaridade (*MPEC*) é um problema de optimização global, em que o conjunto admissível é constituído por um problema complementar ou de desigualdades variacionais. Este problema tem surgido associado a fenómenos de equilíbrio e tem diversas aplicações nas áreas citadas anteriormente [Luo et al., 1997, Outrata et al., 1998, Dirkse et al., 2005]. Tendo aparecido primeiramente nos anos sessenta com o trabalho de K. Kirchgässner [Kirchgässner, 1962], tem sido desde então objecto de intensa investigação. Para além das aplicações referidas, a sua importância advém de muitos problemas de optimização global se poderem resolver eficientemente, explorando as suas equivalências a *MPECs* [Júdice e Faustino, 1991, Júdice e Faustino, 1992]. Neste trabalho apresentamos inicialmente novas formulações de um programa bilinear e linear complementar (*LCP*) como *MPECs* e analisamos a importância dessas formulações *MPEC* como meios para a solução desses problemas.

A procura de técnicas eficientes para a resolução do problema linear complementar tem-se intensificado através da construção de classes de matrizes [Cottle et al., 1992, Murty, 1988]. Em geral o problema linear complementar é um problema NP-difícil e por isso apenas um método enumerativo é capaz de o resolver eficazmente. O método sequencial complementar [Júdice e Faustino, 1992] e o método complementar de pesquisa em árvore [Júdice et al., 2006b] são alguns dos algoritmos enumerativos existentes para a determinação do óptimo global do *MPEC* linear com restrições lineares. Neste trabalho são estudados e comparados estes processos e é desenvolvido, implementado e testado um novo método enumerativo paramétrico para o mesmo objectivo.

A programação inteira é um dos problemas de optimização que mais atenção tem suscitado

dos investigadores, devido à sua enorme variedade de aplicações. A dificuldade de resolução dos problemas inteiros está associada ao facto do conjunto admissível ser discreto. Neste trabalho é abordada uma formulação *MPEC* simples do programa e é provado que a formulação de dois níveis do problema inteiro 0-1 apresentada em [Audet et al., 1997] é um seu caso particular. Problemas inteiros de diversos tipos são resolvidos usando os algoritmos *MPEC* já mencionados.

O problema da determinação do conjunto independente máximo de um grafo é um problema de optimização combinatória NP-completo [Garey e Johnson, 1983], com um vasto leque de aplicações [Du e Pardalos, 1999, Berge, 1982]. O desenvolvimento de algoritmos eficientes de resolução do problema tem sido alvo de muita investigação. Nesta tese são discutidas formulações quadráticas e *MPEC* do problema do conjunto independente máximo. É introduzido um algoritmo de ramificação e limitação para a resolução de um desses programas quadráticos, baseado na decomposição de uma função numa diferença de duas funções convexas (DC). É também analisada a utilidade do método de *Lemke* para a determinação de um conjunto independente através da formulação *MPEC* do problema. A aplicação do método sequencial complementar à resolução desse *MPEC* é também investigada.

O cálculo do número de condição de uma matriz é um dos temas clássicos de análise numérica. Para matrizes de dimensão elevada, o cálculo do valor exacto do número de condição é impraticável, pelo que têm sido utilizados vários processos para a sua estimação. Como última contribuição desta tese salienta-se o desenvolvimento de um estimador do número de condição para a norma l_1 , baseado numa formulação *MPEC* do problema. É ainda mostrado que a estimação do número de condição de uma matriz de *Minkowski* pode ser feita com apenas um sistema de equações lineares.

Na estruturação desta tese, no primeiro capítulo são introduzidos o problema linear complementar e o *MPEC*, as classes de matrizes e as propriedades para a existência de solução para esses problemas. São ainda apresentadas várias formulações de problemas de optimização como problemas *MPEC*.

No segundo capítulo são analisadas algumas técnicas para a determinação de um ponto estacionário ou óptimo global do *MPEC*. São discutidos o método complementar de restrições activas [Júdice et al., 2005], o algoritmo sequencial complementar [Júdice e Faustino, 1992], o método complementar de pesquisa em árvore [Júdice et al., 2006b] e o novo método enumerativo paramétrico.

No terceiro capítulo é abordada a resolução de problemas inteiros binários através das suas formulações *MPEC*. As adaptações dos algoritmos sequencial complementar e enumerativo paramétrico para estes problemas são descritas e é apresentada experiência computacional com problemas binários de vários tipos.

No capítulo quatro é apresentado o algoritmo de ramificação e limitação para uma formulação quadrática do problema da determinação do conjunto independente máximo de um grafo.

São ainda deduzidas formulações *MPEC* para esse objectivo e é feita uma análise da determinação do conjunto independente a partir da resolução desses problemas.

No capítulo cinco é introduzido e analisado o estimador para o cálculo do número de condição de uma matriz relativa à norma l_1 , através da sua formulação como um *MPEC*.

Capítulo 1

Programação Matemática com Restrições de Equilíbrio

Neste capítulo introduzem-se os Problemas Linear Complementar e de Programação Matemática com Restrições de Equilíbrio (*MPEC*) e apresentam-se algumas formulações de problemas de optimização como *MPECs*.

1.1 Problema Linear Complementar

O Problema Linear Complementar (*LCP*) consiste em encontrar vectores $z \in \mathbb{R}^n$ e $w \in \mathbb{R}^n$ tais que

$$\begin{aligned}w &= q + Mz \\ z &\geq 0, w \geq 0 \\ z^T w &= 0,\end{aligned}$$

onde $M \in \mathbb{R}^{n \times n}$ e $q \in \mathbb{R}^n$ são respectivamente uma matriz e vector dados. Deste conjunto de restrições devem distinguir-se as lineares, que definem o conjunto admissível

$$K = \{(z, w) \in \mathbb{R}^{2n} : w = q + Mz, z \geq 0, w \geq 0\} \quad (1.1)$$

e as condições de complementaridade

$$z_i w_i = 0, i = 1, \dots, n \quad (1.2)$$

equivalentes a $z^T w = 0$ para $z \geq 0$ e $w \geq 0$. Tal como em programação linear, uma solução que satisfaça as restrições (1.1) diz-se admissível. Além disso é complementar se satisfizer as restrições $w = q + Mz$ e (1.2), ou seja, se para cada valor de i pelo menos uma das variáveis z_i ou w_i for nula. As variáveis z_i e w_i dizem-se complementares entre si.

Na resolução de um LCP podem surgir quatro casos distintos:

- (i) LCP tem uma única solução.
- (ii) LCP tem várias soluções.
- (iii) LCP é não admissível, isto é, $K = \emptyset$.
- (iv) LCP é admissível ($K \neq \emptyset$) mas não tem solução.

Os principais resultados teóricos relativos à existência e unicidade de soluções do LCP e à complexidade da sua resolução estão normalmente relacionados com classes de matrizes [Cottle et al., 1992, Murty, 1988], que são apresentadas a seguir.

Definição 1.1 *Matrizes Definidas Positivas (PD) e Semi-Definidas Positivas (PSD)*

- $M \in PD \Leftrightarrow z^T M z > 0, \forall z \neq 0$.
- $M \in PSD \Leftrightarrow z^T M z \geq 0, \forall z$.

Matrizes Semi-Definidas Negativas (NSD) e Indefinidas (IND)

- $M \in NSD \Leftrightarrow z^T M z \leq 0, \forall z$.
- $M \in IND \Leftrightarrow M \notin PSD$ e $M \notin NSD$.

Matrizes Copositivas (C) e Copositivas Estritas (SC)

- $M \in SC \Leftrightarrow z^T M z > 0, \forall 0 \neq z \geq 0$.
- $M \in C \Leftrightarrow z^T M z \geq 0, \forall z \geq 0$.

Matrizes P , P_0 , Suficientes (Suf), Suficientes Por Linhas (RSuf) e Suficientes Por Colunas (CSuf)

- $M \in P \Leftrightarrow$ menores principais de M são positivos.
- $M \in P_0 \Leftrightarrow$ menores principais de M são não negativos.
- $M \in CSuf \Leftrightarrow [z_i(Mz)_i \leq 0 \ \forall i \Rightarrow z_i(Mz)_i = 0 \ \forall i]$.
- $M \in RSuf \Leftrightarrow M^T \in CSuf$.
- $M \in Suf \Leftrightarrow M \in RSuf \wedge M \in CSuf$.

As relações entre estas classes de matrizes apresentam-se nos diagramas a seguir, onde o símbolo \subset representa inclusão estrita.

$$\begin{array}{ccc} PSD & \subset & C \\ \cup & & \cup \\ PD & \subset & SC \end{array} \qquad \begin{array}{ccccc} P & \subset & Suf & \subset & RSuf & \subset & P_0 \\ \cup & & \cup & & & & \\ PD & \subset & PSD. & & & & \end{array}$$

O próximo teorema resume os resultados mais importantes relacionados com a existência e unicidade de solução do *LCP* para essas classes [Cottle et al., 1992, Murty, 1988].

Teorema 1.1

- (i) $M \in P$ se e só se o *LCP* tem uma única solução para cada $q \in \mathbb{R}^n$.
- (ii) Se $M \in RSuf$, então o *LCP* tem pelo menos uma solução para cada q tal que $K \neq \emptyset$.
- (iii) Se $M \in SC$, então o *LCP* tem pelo menos uma solução para cada $q \in \mathbb{R}^n$.

Em particular, o resultado (ii) é válido para $M \in PSD$ e além disso o *LCP* tem uma solução única para cada $q \in \mathbb{R}^n$ se $M \in PD$. Notar ainda que para matrizes simétricas,

$$PD = P \quad \text{e} \quad PSD = Suf = P_0.$$

Definição 1.2 Um *LCP* diz-se monótono se $M \in PSD$ e diz-se estritamente monótono se $M \in PD$.

Os *LCPs* monótonos e estritamente monótonos podem ser resolvidos em tempo polinomial [Kojima et al., 1989], enquanto que os não monótonos são em geral *NP*-difíceis [Murty, 1988]. Notar que a classe de complexidade de resolução de um *LCP* com $M \in P$, *Suf* ou *SC* é um problema em aberto. Contudo, o *LCP* é *NP*-difícil se $M \in P_0$, *IND* ou *NSD* [Murty, 1988, Kojima et al., 1989].

Como veremos mais adiante, um dos processos mais usados para resolver o *LCP* consiste em procurar determinar a solução óptima do seguinte Programa Quadrático (*QP*)

$$\begin{array}{ll} \text{Minimizar} & z^T w \\ \text{sujeito a} & (z, w) \in K, \end{array}$$

com K o conjunto admissível do *LCP*. Como

$$z^T w = z^T (q + Mz) = q^T z + \frac{1}{2} z^T (M + M^T) z$$

podemos escrever esse programa na forma

$$\begin{aligned} QP : \quad & \text{Minimizar} \quad f(z) = q^T z + \frac{1}{2} z^T (M + M^T) z \\ & \text{sujeito a} \quad Mz \geq -q \\ & \quad \quad \quad z \geq 0. \end{aligned}$$

Como a função quadrática é limitada inferiormente na região definida pelas restrições do QP quando $K \neq \emptyset$, então esse programa tem solução óptima [Murty, 1988] e três casos podem acontecer:

- (i) $K = \{z \in \mathbb{R}^n : Mz \geq -q, z \geq 0\} = \emptyset$ e o LCP é não admissível.
- (ii) $K \neq \emptyset$ e $f(\bar{z}) = \min_{z \in K} f(z) = 0$ e \bar{z} é solução do LCP .
- (iii) $K \neq \emptyset$ e $\min_{z \in K} f(z) > 0$ e o LCP é admissível mas não tem solução.

Definição 1.3 Um ponto $\bar{z} \in K$ é um Ponto Estacionário de f em K se satisfizer as condições de Karush-Kuhn-Tucker associadas ao programa QP , isto é, se existirem vectores $w \in \mathbb{R}^n$ e $u \in \mathbb{R}^n$ tais que

$$\begin{aligned} w &= q + (M + M^T)z - M^T u \\ Mz &\geq -q \\ z &\geq 0, w \geq 0, u \geq 0 \\ z^T w &= 0 \\ u^T (q + Mz) &= 0. \end{aligned}$$

Atendendo a esta definição, verifica-se o seguinte resultado [Cottle et al., 1992].

Teorema 1.2 Se $M \in RSuf$, então todo o ponto estacionário de QP é solução do LCP .

Em particular, este resultado é verdadeiro para matrizes P e PSD . Portanto para $LCPs$ monótonos ou quando a matriz M é $RSuf$, o LCP pode ser resolvido a partir da determinação de um ponto estacionário do QP . Notar que para $M \in RSuf$, o caso (iii) apresentado anteriormente não pode acontecer, o que está de acordo com o Teorema 1.1. Além disso se $M \in P$, então $K \neq \emptyset$ e apenas o segundo caso (ii) pode ocorrer. A dificuldade de resolução do LCP está de algum modo relacionada com este teorema e os problemas difíceis são aqueles em que um ponto estacionário do QP não fornece uma solução do LCP . Como veremos mais adiante, é ainda mais difícil resolver $LCPs$ admissíveis que não tenham solução.

1.2 Problema de Programação Matemática com Restrições de Equilíbrio (*MPEC*)

O Problema de Programação Matemática com Restrições de Equilíbrio ou de Complementaridade, do inglês, *Mathematical Programming with Equilibrium (Complementarity) Constraints (MPEC)* é um problema de optimização cujas restrições incluem problemas complementares ou de desigualdades variacionais paramétricas. Estas restrições estão tipicamente relacionadas com fenómenos de equilíbrio, que surgem em áreas como a engenharia e a economia [Luo et al., 1997, Outrata et al., 1998, Dirkse et al., 2005]. Para além deste facto, a importância deste problema advém de muitos problemas de optimização global se poderem resolver eficientemente explorando a sua equivalência a *MPECs* [Júdice e Faustino, 1991, Júdice e Faustino, 1992]. Estes problemas surgiram primeiramente nos anos sessenta com o trabalho de K. Kirchgässner [Kirchgässner, 1962] e desde então têm sido objecto de intensa investigação. Estudos detalhados deste tipo de problema são apresentados em [Luo et al., 1997, Outrata et al., 1998, Dirkse et al., 2005].

Nesta tese iremos considerar apenas o caso de ser linear o problema complementar que constitui as restrições do *MPEC*. O problema pode então ser definido da seguinte forma

$$\begin{aligned} & \text{Minimizar} && f(z, y) \\ & \text{sujeito a} && Ew = q + Mz + Ny \\ & && w \geq 0, z \geq 0, y \in K_y \\ & && z^T w = 0, \end{aligned}$$

onde $E \in \mathbb{R}^{l \times n}$, $M \in \mathbb{R}^{l \times n}$, $N \in \mathbb{R}^{l \times m}$, $z, w \in \mathbb{R}^n$, $q \in \mathbb{R}^l$, $K_y \subseteq \mathbb{R}^m$ é um poliedro convexo em y e $f : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$ é continuamente diferenciável em $\mathbb{R}^{n \times m}$. O conjunto de restrições do *MPEC* é um Problema Complementar Generalizado, que será denotado por *GLCP*, do inglês *Generalized Complementarity Problem*. Da definição do problema constata-se que o *GLCP* é um Problema Linear Complementar paramétrico com um número n de parâmetros. Problemas paramétricos em optimização são normalmente muito difíceis de resolver, o que tem levado os investigadores da complementaridade a procurar outras formas de resolver o *GLCP*. À semelhança do *LCP*, podemos associar ao *GLCP* o seguinte programa quadrático

$$\begin{aligned} QP : \quad & \text{Minimizar} && z^T w = f(z, w) \\ & \text{sujeito a} && Ew = q + Mz + Ny \\ & && y \in K_y \\ & && w \geq 0, z \geq 0. \end{aligned} \tag{1.3}$$

Como veremos mais adiante, em muitas das formulações dos problemas de optimização como

MPECs, o *GLCP* tem a seguinte forma

$$\begin{aligned} \begin{bmatrix} I \\ 0 \end{bmatrix} w &= \begin{bmatrix} q \\ -d \end{bmatrix} + \begin{bmatrix} H & -C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} z \\ u \end{bmatrix} + \begin{bmatrix} R \\ S \end{bmatrix} y \\ Ay &= b, y \geq 0 \\ z &\geq 0, w \geq 0 \\ z^T w &= 0 \end{aligned} \quad (1.4)$$

com $H \in PSD$. Neste caso o programa quadrático (1.3) tem a seguinte forma

$$\begin{aligned} \text{Minimizar} \quad & q^T z + \frac{1}{2} z^T (H + H^T) z - z^T C^T u + z^T R y \\ \text{sujeito a} \quad & H z - C^T u + R y \geq -q \\ & C z + S y = d \\ & A y = b \\ & z \geq 0, y \geq 0 \end{aligned} \quad (1.5)$$

e é possível estender o resultado apresentado em [Júdice e Vicente, 1994] de que todo o ponto estacionário do programa quadrático é uma solução do *GLCP*.

Teorema 1.3 *Se $(\bar{z}, \bar{u}, \bar{y})$ é um ponto estacionário do programa quadrático (1.5) com $H \in PSD$, então $(\bar{z}, \bar{u}, \bar{y})$ é solução do *GLCP*.*

Demonstração: Se $(\bar{z}, \bar{u}, \bar{y})$ é um ponto estacionário do programa quadrático (1.5), então existem multiplicadores $\bar{\alpha}, \bar{\beta}, \bar{\sigma}, \bar{\gamma}, \bar{\theta}$, tais que $(\bar{z}, \bar{u}, \bar{y}, \bar{\alpha}, \bar{\beta}, \bar{\sigma}, \bar{\gamma}, \bar{\theta})$ satisfazem as condições de *Karush-Kuhn-Tucker*

$$\begin{aligned} q + (H + H^T)z - C^T u + R y &= H^T \alpha + C^T \beta + \gamma \\ -C z &= -C \alpha \\ R^T z &= R^T \alpha + S^T \beta + A^T \sigma + \theta \\ q + H z - C^T u + R y &\geq 0 \\ C z + S y &= d \\ A y &= b \\ z \geq 0, y \geq 0, \alpha \geq 0, \gamma \geq 0, \theta \geq 0 \\ \gamma^T z &= \theta^T y = 0 \\ \alpha^T (q + H z - C^T u + R y) &= 0. \end{aligned} \quad (1.6)$$

Multiplicando escalarmente a primeira das equações (1.6) por z e α , obtêm-se as seguintes igualdades

$$z^T (q + H z - C^T u + R y) = z^T H^T (\alpha - z) + \beta^T C z$$

$$\alpha^T(q + Hz - C^T u + Ry) = \alpha^T H^T(\alpha - z) + \beta^T C\alpha + \alpha^T \gamma.$$

Portanto

$$(\alpha - z)^T(q + Hz - C^T u + Ry) = (\alpha - z)^T H(\alpha - z) + \beta^T(C\alpha - Cz) + \alpha^T \gamma.$$

Donde

$$(\alpha - z)^T H(\alpha - z) = -[z^T(q + Hz - C^T u + Ry) + \alpha^T \gamma] \leq 0. \quad (1.7)$$

Como $H \in PSD$, então

$$(\alpha - z)^T H(\alpha - z) = 0.$$

Portanto de (1.7), vem

$$z^T(q + Hz - C^T u + Ry) = \alpha^T \gamma = 0,$$

o que demonstra o resultado pretendido. \square

Devido a este teorema, o *GLCP* pode ser resolvido por um algoritmo local de programação quadrática, havendo dois casos a considerar:

- (i) O programa é não admissível e o *GLCP* é não admissível.
- (ii) Um ponto estacionário é obtido e fornece uma solução do *GLCP*.

Deste modo também se estabelece que todo o *GLCP* da forma (1.4) com $H \in PSD$ tem solução desde que seja admissível. Em geral, este resultado não é verdadeiro e, tal como para o *LCP*, existem *GLCPs* admissíveis sem solução. Além disso o *GLCP* é um problema NP-difícil, pois o mesmo acontece com o *LCP*.

1.3 Formulação *MPEC* do Problema de Dois Níveis

O problema de programação de dois níveis define-se pela existência de uma hierarquia entre dois problemas de optimização, sendo as restrições de um deles, chamado de nível superior ou de primeiro nível, definidas em parte por um segundo problema de optimização paramétrico, denominado de nível inferior ou de segundo nível.

Portanto o programa de dois níveis (*BPP*) na sua forma mais geral tem a seguinte estrutura

$$\begin{aligned} BPP : \quad & \text{Minimizar} \quad f(z, y) \\ & \text{sujeito a} \quad y \in K_y \\ & \quad \quad \quad G_1(z, y) \leq 0 \\ & \quad \quad \quad G_2(z, y) = 0 \\ & \quad \quad \quad z \in \arg \min \{g(z, y) : z \in K_z, H_1(z, y) \leq 0, H_2(z, y) = 0\}, \end{aligned}$$

em que K_z e K_y são subconjuntos poliédricos de \mathbb{R}^n e \mathbb{R}^m , respectivamente, f e g são funções reais tais que $f, g : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$, G_i e H_i , $i = 1, 2$ são funções vectoriais reais, tais que $G_i : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^{l_i}$ e $H_i : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^{p_i}$ com $n, m \in \mathbb{N}$ e $l_i, p_i \in \mathbb{N} \cup \{0\}$, $i = 1, 2$.

As diferentes classes do *BPP* diferem entre si através das particularizações das funções f , g , G_i e H_i e dos conjuntos K_z e K_y . Em todas as classes podem considerar-se programas de dois níveis com ou sem restrições de primeiro nível. O programa do segundo nível é convexo se são convexas a função g e as que constituem a função vectorial H_1 e H_2 é uma função linear. Nesse caso, com uma restrição de qualificação adequada para a função H_1 e se todas as funções do segundo nível são diferenciáveis, é possível substituir o programa do segundo nível pelas suas condições de optimalidade de *Karush-Kuhn-Tucker* e obter um *MPEC*. Como nesta tese pretendemos que todas as restrições do *MPEC* com excepção da condição de complementaridade sejam lineares, então devemos assumir que, além das hipóteses anteriores, G_i e H_1 sejam lineares e g quadrática. Desse modo obtemos o seguinte programa de dois níveis:

$$\begin{aligned} & \text{Minimizar} && f(z, y) \\ & \text{sujeito a} && Cz + Dy = r \\ & && y \in K_y, \end{aligned}$$

onde z é a solução óptima do programa quadrático convexo

$$\begin{aligned} & \text{Minimizar} && c^T z + y^T R z + \frac{1}{2} z^T Q z \\ & \text{sujeito a} && Az + By = b \\ & && z \geq 0. \end{aligned}$$

Substituindo o programa quadrático do segundo nível pelas suas condições de *Karush-Kuhn-Tucker* obtém-se o *MPEC*

$$\begin{aligned} & \text{Minimizar} && f(z, y) \\ & \text{sujeito a} && c + Qz + R^T y = A^T u + w \\ & && Az + By = b \\ & && Cz + Dy = r \\ & && z \geq 0, w \geq 0, y \in K_y \\ & && z^T w = 0. \end{aligned}$$

Notar que se as restrições do primeiro nível $Cz + Dy = r$ não existirem, então o *GLCP* que constitui o conjunto admissível do *MPEC* tem a forma

$$\begin{aligned} \begin{bmatrix} I \\ 0 \end{bmatrix} w &= \begin{bmatrix} c \\ -b \end{bmatrix} + \begin{bmatrix} Q & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} z \\ u \end{bmatrix} + \begin{bmatrix} R^T \\ B \end{bmatrix} y \\ & z \geq 0, w \geq 0, y \in K_y \\ & z^T w = 0. \end{aligned}$$

Como $Q \in PSD$, então o mesmo acontece à matriz [Cottle et al., 1992]

$$\begin{bmatrix} Q & -A^T \\ A & 0 \end{bmatrix}$$

e portanto o *GLCP* tem a forma apresentada em (1.4) se $K_y = \{y : Ay = b, y \geq 0\}$. Assim, se o problema do primeiro nível contiver apenas restrições nas variáveis y , o programa de dois níveis reduz-se a um *MPEC*, cujo conjunto de restrições que formam o *GLCP* tem solução desde que seja admissível. Além disso uma solução do *GLCP* pode ser obtida determinando um ponto estacionário de um programa quadrático, que minimiza a complementaridade no conjunto das restrições lineares do *GLCP*.

1.4 Formulação *MPEC* de Programação Bilinear

Tal como o problema de optimização anterior, a Programação Bilinear tem vindo a ser bastante estudada devido ao seu elevado número de aplicações [Konno, 1971]. A sua definição mais geral inclui uma função bilinear das variáveis z_i e y_i que é minimizada num conjunto convexo definido por restrições lineares. Portanto podemos escrever um programa bilinear na forma

$$\begin{aligned} \text{Minimizar} \quad & f(z, y) = c^T z + d^T y + z^T H y \\ \text{sujeito a} \quad & Az + By = b \\ & z \in K_z, \quad y \in K_y, \end{aligned}$$

onde $K_z \in \mathbb{R}^n$ e $K_y \in \mathbb{R}^m$ são conjuntos poliédricos nas variáveis z e y respectivamente. Se considerarmos

$$K_z = \{z \in \mathbb{R}^n : Cz = g, z \geq 0\}$$

então podemos escrever o programa bilinear na forma

$$\begin{aligned} \text{Minimizar} \quad & d^T y + \min_z \{(c + Hy)^T z : Az + By = b, Cz = g, z \geq 0\}. \\ & y \in K_y \end{aligned} \tag{1.8}$$

Consideremos agora o programa linear interior nas variáveis z_i

$$\begin{aligned} \text{Minimizar} \quad & (c + Hy)^T z \\ \text{sujeito a} \quad & Az = b - By \\ & Cz = g \\ & z \geq 0. \end{aligned} \tag{1.9}$$

O seu dual é definido por

$$\begin{aligned} \text{Maximizar} \quad & (b - By)^T u + g^T v \\ \text{sujeito a} \quad & A^T u + C^T v \leq c + Hy. \end{aligned} \tag{1.10}$$

Se agora introduzirmos as variáveis de folga w_i associadas às restrições de desigualdade do dual e tendo em conta o teorema da complementaridade das variáveis de folga [Murty, 1988], então o programa interior (1.9) pode ser substituído por um *MPEC* que contém as restrições do primal (1.9) e do dual (1.10) e a restrição de complementaridade das variáveis de folga. Substituindo esse problema em (1.8), obtemos o seguinte *MPEC*

$$\begin{aligned}
 & \text{Minimizar} \quad d^T y + g^T v + b^T u - u^T B y \\
 & \text{sujeito a} \quad \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix} w = \begin{bmatrix} c \\ -b \\ -g \end{bmatrix} + \begin{bmatrix} 0 & -A^T & -C^T \\ A & 0 & 0 \\ C & 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ u \\ v \end{bmatrix} + \begin{bmatrix} H \\ B \\ 0 \end{bmatrix} y \\
 & \quad \quad \quad z \geq 0, w \geq 0, y \in K_y \\
 & \quad \quad \quad z^T w = 0.
 \end{aligned}$$

Como a matriz

$$\begin{bmatrix} 0 & -A^T & -C^T \\ A & 0 & 0 \\ C & 0 & 0 \end{bmatrix}$$

é *PSD*, então mais uma vez o *GLCP* que constitui as restrições do *MPEC* tem a forma (1.4). Portanto a determinação de uma solução do conjunto *GLCP* de restrições do *MPEC* pode ser feita determinando um ponto estacionário de um programa quadrático que tem a complementaridade como função e as restrições lineares do *GLCP*.

Um Programa Bilinear diz-se separado se não contiver restrições nos dois tipos de variáveis z e y , isto é, se o seu conjunto de restrições é $K_z \times K_y$. Nesse caso o *MPEC* tem a forma

$$\begin{aligned}
 & \text{Minimizar} \quad d^T y + g^T v \\
 & \text{sujeito a} \quad \begin{bmatrix} I \\ 0 \end{bmatrix} w = \begin{bmatrix} c \\ -g \end{bmatrix} + \begin{bmatrix} 0 & -C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} z \\ v \end{bmatrix} + \begin{bmatrix} H \\ 0 \end{bmatrix} y \\
 & \quad \quad \quad z \geq 0, w \geq 0, y \in K_y \\
 & \quad \quad \quad z^T w = 0.
 \end{aligned}$$

Portanto a função é neste caso linear, o que torna mais fácil o processo de resolução de um programa bilinear através do *MPEC* [Júdice e Faustino, 1991]. É ainda importante notar que esta equivalência entre Programas Bilineares e *MPECs* só é verdadeira se o programa interior tiver solução ótima. Tal é evidentemente assegurado se o conjunto K_z é limitado, o que é uma hipótese normalmente considerada na determinação do mínimo global de um programa bilinear [Horst et al., 2000].

1.5 Formulações *MPEC* do Problema Linear Complementar

Consideremos novamente o Problema Linear Complementar (*LCP*)

$$\begin{aligned} w &= q + Mz \\ z &\geq 0, w \geq 0 \\ z^T w &= 0. \end{aligned}$$

Uma das abordagens mais usuais para o tratamento do *LCP* consiste em transformá-lo num problema complementar alargado [Cottle et al., 1992]. Uma das formas de o fazer considera o problema linear complementar

$$\begin{aligned} (ALCP) \quad \begin{bmatrix} v \\ w \end{bmatrix} &= \begin{bmatrix} e \\ q \end{bmatrix} + \begin{bmatrix} 0 & -I \\ M & 0 \end{bmatrix} \begin{bmatrix} z \\ u \end{bmatrix} \\ v, w, z, u &\geq 0 \\ z^T v &= w^T u = 0, \end{aligned}$$

com $e \in \mathbb{R}^n$ um vector de componentes unitárias. O próximo teorema mostra que os dois problemas são equivalentes.

Teorema 1.4 *LCP e ALCP são equivalentes.*

Demonstração: Se (\bar{z}, \bar{w}) é solução do *LCP*, seja

$$I = \{i \in \{1, \dots, n\} : \bar{z}_i > 0\}.$$

Então $\bar{w}_i = 0$ para todo o $i \in I$ e além disso $\bar{z}_j = 0, \bar{w}_j \geq 0$ para todo $j \notin I$. Consideremos agora os vectores \bar{u} e \bar{v} definidos por

$$\bar{u}_i = \begin{cases} 1, & \text{se } i \in I \\ 0, & \text{se } i \notin I \end{cases}, \quad \bar{v}_i = \begin{cases} 0, & \text{se } i \in I \\ 1, & \text{se } i \notin I. \end{cases}$$

Então $(\bar{z}, \bar{w}, \bar{u}, \bar{v})$ é solução do *ALCP*.

Para estabelecer a segunda parte, seja $(\bar{z}, \bar{w}, \bar{u}, \bar{v})$ uma solução do *ALCP*. Então $\bar{w} = q + M\bar{z}$, $\bar{z} \geq 0$, $\bar{w} \geq 0$ e portanto (\bar{z}, \bar{w}) é solução do *LCP* desde que $\bar{w}_i = 0$ para todo o i tal que $\bar{z}_i > 0$. Mas pela complementaridade do *ALCP* vem

$$\bar{z}_i > 0 \Rightarrow \bar{v}_i = 0 \Rightarrow \bar{u}_i = 1 \Rightarrow \bar{w}_i = 0,$$

o que demonstra o resultado. □

Consideremos agora o *ALCP*. Tal como para o *LCP*, uma forma de o resolver consiste em minimizar a complementaridade $(z^T v + w^T u)$ no conjunto das restrições lineares do *ALCP*.

Mas

$$\begin{aligned} z^T v + w^T u &= z^T(e - u) + u^T(q + Mz) = \\ &= e^T z + q^T u + u^T(M - I)z, \end{aligned}$$

que é uma função bilinear nas variáveis z_i e u_i . Portanto o *ALCP* reduz-se a um Problema Bilinear Separado

$$\begin{aligned} \text{Minimizar} \quad & e^T z + q^T u + u^T(M - I)z \\ \text{sujeito a} \quad & Mz \geq -q, \quad z \geq 0 \\ & 0 \leq u \leq e. \end{aligned}$$

Como vimos anteriormente, este programa é equivalente a um de dois *MPECs* lineares que diferem entre si na escolha da minimização interiormente em relação às variáveis z_i e u_j , respectivamente. Se considerarmos a minimização em z nesse programa, obtemos o seguinte *MPEC*

$$\begin{aligned} \text{Minimizar} \quad & q^T u - q^T v \\ \text{sujeito a} \quad & \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} e \\ q \end{bmatrix} + \begin{bmatrix} 0 & -M^T \\ M & 0 \end{bmatrix} \begin{bmatrix} z \\ v \end{bmatrix} + \begin{bmatrix} M^T - I \\ 0 \end{bmatrix} u \\ & \begin{bmatrix} z \\ v \end{bmatrix}, \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \geq 0, \quad \begin{bmatrix} z \\ v \end{bmatrix}^T \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = 0 \\ & 0 \leq u \leq e. \end{aligned} \tag{1.11}$$

Como é evidente, o *GLCP* obtido tem a forma (1.4) e portanto pode ser resolvido através da determinação de um ponto estacionário de um programa quadrático. Notar ainda que o valor ótimo do *MPEC* é igual a zero, o que, como veremos mais adiante, tem importantes implicações na resolução do *LCP* através de técnicas globais para o *MPEC*. Para esse efeito, é muitas vezes importante determinar a solução ótima do programa relaxado que se obtém do *MPEC* por supressão da condição de complementaridade, isto é, do problema

$$\begin{aligned} (LP) \quad \text{Minimizar} \quad & q^T u - q^T v = f(z, u, v) \\ \text{sujeito a} \quad & (M^T - I)u - M^T v \geq -e \\ & Mz \geq -q \\ & 0 \leq u \leq e \\ & z \geq 0, \quad v \geq 0. \end{aligned}$$

O próximo teorema mostra que este programa é ilimitado inferiormente, desde que o conjunto admissível do *LCP* seja limitado. Nesse teorema iremos assumir que o *LCP* é não trivial, isto é, que existe pelo menos uma componente negativa do vector q .

Teorema 1.5 *Se o LCP é não trivial e o seu conjunto admissível*

$$K = \{z \in \mathbb{R}^n : Mz \geq -q, z \geq 0\}$$

é limitado e não vazio, então $f(z, u, v)$ tende para $-\infty$ no conjunto admissível do programa linear relaxado (LP).

Demonstração: Dado um programa linear

$$\begin{aligned} & \text{Minimizar} && c^T x \\ & \text{sujeito a} && Ax \geq b \\ & && x \geq 0, \end{aligned}$$

então $c^T x$ tende para $-\infty$ no seu conjunto admissível se e só se existe um vector y satisfazendo $Ay \geq 0$, $y \geq 0$, $c^T y < 0$ [Murty, 1983]. Portanto para (LP) é necessário encontrar um vector $(\bar{z}, \bar{u}, \bar{v})$ satisfazendo

$$\begin{aligned} (M^T - I)u - M^T v &\geq 0 \\ Mz &\geq 0 \\ -u &\geq 0 \\ z \geq 0, u \geq 0, v &\geq 0 \\ q^T u - q^T v &< 0. \end{aligned} \tag{1.12}$$

A terceira desigualdade implica que $\bar{u} = 0$. Além disso $\bar{z} = 0$ satisfaz a segunda igualdade. Portanto o teorema fica provado se se encontrar um vector \bar{v} tal que

$$v \geq 0, M^T v \leq 0, q^T v > 0. \tag{1.13}$$

Para esse fim consideremos o programa linear

$$\begin{aligned} & \text{Minimizar} && -e^T x \\ & \text{sujeito a} && Mx \geq -q \\ & && x \geq 0 \end{aligned} \tag{1.14}$$

com e um vector de componentes unitárias. Como o seu conjunto admissível é não vazio e limitado, então o programa tem uma solução óptima. Por outro lado, para que o LCP não seja trivial, tem de existir pelo menos uma componente negativa no vector q . Então o vector nulo não pertence ao conjunto admissível e o valor óptimo é negativo. Consideremos agora o seu dual

$$\begin{aligned} & \text{Maximizar} && -q^T v \\ & \text{sujeito a} && M^T v \leq -e \\ & && v \geq 0. \end{aligned}$$

Da teoria da Dualidade Linear, este programa tem também uma solução óptima \bar{v} de valor negativo que por isso satisfaz as condições (1.13). Portanto $(\bar{u} = 0, \bar{z} = 0, v = \bar{v})$ é solução do sistema de desigualdades (1.12) e o programa linear relaxado (LP) é ilimitado inferiormente.

□

Notar que o teorema mantém-se verdadeiro no caso do conjunto admissível do programa linear (1.14) não ser limitado mas a sua função linear ser limitada inferiormente nesse conjunto.

A formulação do *LCP* como um *MPEC* tem dois inconvenientes, a saber, a duplicação do número de variáveis e o facto do programa linear relaxado não ter solução óptima. No capítulo 2 iremos discutir a importância desses factores na resolução do *LCP* através do uso de técnicas globais para o *MPEC*.

Seguidamente apresentamos uma outra formulação do *LCP* como *MPEC* cujo programa relaxado tem solução óptima. Para isso consideremos a formulação quadrática do *LCP* referida anteriormente

$$\begin{aligned} \text{Minimizar} \quad & f(z) = q^T z + \frac{1}{2} z^T (M + M^T) z \\ \text{sujeito a} \quad & Mz \geq -q \\ & z \geq 0. \end{aligned}$$

Um ponto estacionário desse programa satisfaz as seguintes condições de *Karush-Kuhn-Tucker*

$$\begin{aligned} q + (M + M^T)z &= M^T \alpha + w \\ Mz &\geq -q \\ z \geq 0, w \geq 0, \alpha \geq 0 \\ z^T w &= 0 \\ \alpha^T (Mz + q) &= 0. \end{aligned}$$

É então fácil concluir que (z, w) é uma solução do *LCP* se (α, z, w) é solução óptima de valor nulo do seguinte *MPEC* quadrático com restrições lineares

$$\begin{aligned} \text{Minimizar} \quad & (\alpha - z)^T (\alpha - z) \\ \text{sujeito a} \quad & w = q + (M + M^T)z - M^T \alpha \\ & z \geq 0, \alpha \geq 0, w \geq 0 \\ & z^T w = 0. \end{aligned}$$

Esta formulação *MPEC* tem duas vantagens relativamente à anterior, nomeadamente, o seu programa relaxado ter uma solução óptima e o número de restrições lineares de igualdade ser exactamente igual ao do *LCP*. Por outro lado, a determinação de uma solução do *GLCP* é obtida facilmente a partir da determinação de um ponto estacionário do programa quadrático (*QP*) associado ao *LCP*. As desvantagens desta formulação dizem respeito ao facto da matriz $(M + M^T)$ não ser semi-definida positiva (se o fosse então não havia necessidade de considerar o *MPEC*) e a função objectivo ser quadrática. Esta última desvantagem pode ser ultrapassada usando a norma l_1 em vez da l_2 na definição do *MPEC*, isto é, considerar

o seguinte programa não linear

$$\begin{aligned} & \text{Minimizar} && \sum_{i=1}^n |\alpha_i - z_i| \\ & \text{sujeito a} && w = q + (M + M^T)z - M^T\alpha \\ & && z \geq 0, \alpha \geq 0, w \geq 0 \\ & && z^T w = 0. \end{aligned}$$

Para evitar o uso dos módulos, consideremos para cada $i = 1, \dots, n$ as substituições

$$\begin{aligned} \alpha_i - z_i &= u_i - v_i \\ u_i &\geq 0, v_i \geq 0 \\ u_i v_i &= 0. \end{aligned} \tag{1.15}$$

Então

$$|\alpha_i - z_i| = u_i + v_i$$

para todo o $i = 1, \dots, n$ e podemos escrever esse programa como o seguinte *MPEC*

$$\begin{aligned} & \text{Minimizar} && e^T u + e^T v = f(z, u, v) \\ & \text{sujeito a} && w = q + Mz - M^T u + M^T v \\ & && z + u - v \geq 0 \\ & && z \geq 0, w \geq 0, u \geq 0, v \geq 0 \\ & && z^T w = 0, u^T v = 0. \end{aligned}$$

Seguidamente vamos provar que a restrição $u^T v = 0$ é redundante. Com efeito, se resolvermos o *MPEC* sem essa restrição e se a sua solução óptima $(\bar{z}, \bar{u}, \bar{v})$ não a satisfizer, então tem de existir i tal que $\bar{u}_i > 0$ e $\bar{v}_i > 0$. Seja

$$\theta = \min\{\bar{u}_i, \bar{v}_i\} > 0$$

e considere-se e^i um vector com todas as componentes nulas com excepção da i que tem valor unitário. Então a solução

$$\tilde{z} = \bar{z}, \quad \tilde{u} = \bar{u} - \theta e^i, \quad \tilde{v} = \bar{v} - \theta e^i$$

é admissível para o programa dado e além disso satisfaz

$$\begin{aligned} \tilde{u}_i \tilde{v}_i &= 0 \\ f(\tilde{z}, \tilde{u}, \tilde{v}) &= f(\bar{z}, \bar{u}, \bar{v}) - 2\theta. \end{aligned}$$

Portanto $(\bar{z}, \bar{u}, \bar{v})$ não pode ser solução óptima do *MPEC*. Então a restrição $u^T v = 0$ tem de ser verificada pela solução óptima do *MPEC*, que assim se pode escrever na forma

$$\begin{aligned} & \text{Minimizar} && e^T u + e^T v \\ & \text{sujeito a} && w = q + Mz - M^T u + M^T v \\ & && z + u - v \geq 0 \\ & && z \geq 0, w \geq 0, u \geq 0, v \geq 0 \\ & && z^T w = 0. \end{aligned} \tag{1.16}$$

Como anteriormente, (\bar{z}, \bar{w}) é uma solução do *LCP* se $(\bar{z}, \bar{w}, \bar{u}, \bar{v})$ é uma solução óptima do *MPEC* (1.16) de valor óptimo nulo.

À semelhança da formulação (1.11), este *MPEC* tem os inconvenientes de duplicar o número de variáveis. Além disso a matriz correspondente às variáveis z não é quadrada e por isso não é *PSD*. Como vantagens desta formulação salienta-se o facto de ser fácil determinar uma solução do *GLCP* a partir do ponto estacionário do programa quadrático associado ao *LCP* e das fórmulas (1.15), da função do *MPEC* ser linear e da solução óptima do programa linear relaxado existir. No próximo capítulo iremos discutir a possibilidade de resolver o *LCP* a partir das suas formulações como Problemas de Programação Matemática com Restrições de Equilíbrio.

Capítulo 2

Algoritmos para Programação Matemática com Restrições Lineares de Equilíbrio

Neste capítulo iremos apresentar os algoritmos mais eficientes para a determinação de um ponto estacionário e de um mínimo global para um *MPEC* com restrições lineares e introduzimos um novo algoritmo para o último objectivo.

2.1 Algoritmo Enumerativo para o *GLCP*

Como vimos no capítulo anterior, o *MPEC* na sua forma mais geral é definido por

$$\begin{aligned} \text{Minimizar} \quad & f(w, z, y) \\ \text{sujeito a} \quad & Ew = q + Mz + Ny \\ & w \geq 0, z \geq 0, y \in K_y \\ & z^T w = 0, \end{aligned} \tag{2.1}$$

onde $E \in \mathbb{R}^{l \times n}$, $M \in \mathbb{R}^{l \times n}$, $N \in \mathbb{R}^{l \times m}$, $z, w \in \mathbb{R}^n$, $q \in \mathbb{R}^l$, $f : \mathbb{R}^{2n \times m} \rightarrow \mathbb{R}$ é continuamente diferenciável em $\mathbb{R}^{2n \times m}$ e $K_y \subseteq \mathbb{R}^m$ é um poliedro convexo em y . O seu conjunto de restrições é um problema complementar generalizado denotado por *GLCP*. Em algumas das formulações de problemas de optimização e em muitas aplicações o *GLCP* tem uma estrutura especial, que permite a sua resolução de um modo muito eficiente. É o caso, por exemplo, dos *GLCPs* associados a algumas das formulações discutidas no capítulo anterior. Em geral o *GLCP* é um problema NP-difícil e por isso só um método enumerativo é capaz de o resolver eficazmente.

A ideia do algoritmo enumerativo [Al-Khayyal, 1987, Júdice e Faustino, 1988] pode ser visualizada pelo habitual processo de ramificação em árvore binária, definida pela dicotomia $z_i = 0$ ou $w_i = 0$ presente numa solução complementar (Figura 2.1). Dado que as restrições não complementares do problema (2.1) são todas lineares, então essa pesquisa termina num número finito de nós.

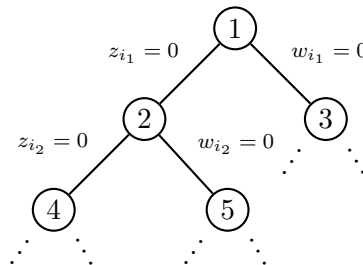


Figura 2.1: Ramificação no método enumerativo

Em cada nó da árvore é determinado um ponto estacionário de um programa quadrático da forma

$$\begin{aligned}
 & \text{Minimizar} && z^T w \\
 & \text{sujeito a} && Ew = q + Mz + Ny \\
 & && z_i = 0, \quad i \in I \\
 & && w_j = 0, \quad j \in J \\
 & && w \geq 0, \quad z \geq 0, \quad y \in K_y,
 \end{aligned} \tag{2.2}$$

com I e J os conjuntos definidos por

$$I = \{i : z_i = 0 \text{ fixa}\}$$

$$J = \{i : w_i = 0 \text{ fixa}\},$$

no caminho da árvore desse nó até à raiz. Como é usual em técnicas enumerativas, o algoritmo usa regras heurísticas para a escolha do nó a ramificar em cada iteração e para a selecção do par de variáveis a ramificar nesse nó, que se definem a seguir.

- Escolha do nó a ramificar: aquele que apresenta o valor mínimo para $z^T w$.
- Escolha do par de variáveis complementares: aquele que apresenta o valor máximo para o produto $z_r w_r$.

Na resolução de um *GLCP* com solução, o algoritmo termina assim que obtiver um nó cujo ponto estacionário associado tiver valor nulo. Esse critério de paragem permite normalmente que o processo termine precocemente com uma pesquisa relativamente reduzida. Como referimos no capítulo anterior, o *GLCP* pode ser admissível e não ter solução. Nesse caso o

algoritmo tem de fazer uma pesquisa exaustiva na árvore antes de terminar. Como veremos mais adiante, estas características do algoritmo tornam-no particularmente recomendado para a resolução de *GLCPs* com várias soluções básicas, menos eficiente para *GLCPs* com um pequeno número de soluções básicas e muito pouco eficaz para lidar com *GLCPs* admissíveis que não tenham solução. É ainda de notar que em alguns casos importantes o algoritmo pode terminar no nó raiz sem efectuar qualquer tipo de pesquisa na árvore. Esse é por exemplo o caso dos *GLCPs* da forma (1.4) que, como vimos anteriormente, aparecem frequentemente em formulações de problemas de optimização não convexos como *MPECs*.

Os passos do algoritmo apresentam-se seguidamente.

ALGORITMO ENUMERATIVO

Passo 0: *Inicialização*

Seja $\mathcal{L} = \{1\}$ a lista inicial de nós em aberto e $QP(1)$ o programa quadrático (2.2) associado a $I_1 = J_1 = \emptyset$. Faça $k = 1$.

Passo 1: *Avaliação de Solução*

Se $\mathcal{L} = \emptyset$ pare: o *GLCP* não tem solução. Caso contrário, escolha um nó $t \in \mathcal{L}$ de acordo com a heurística acima referida e faça $\mathcal{L} = \mathcal{L} \setminus \{t\}$.

Passo 2: *Determinação de Ponto Estacionário*

Aplique um algoritmo de restrições activas ao programa quadrático $QP(t)$ associado ao nó t . Se $QP(t)$ é não admissível volte para o Passo 1. Caso contrário seja $(\bar{w}, \bar{z}, \bar{y})$ o ponto estacionário do problema $QP(t)$ obtido.

Passo 3: *Critério de Paragem*

Se $\bar{z}^T \bar{w} = 0$ pare: $(\bar{w}, \bar{z}, \bar{y})$ é solução do *GLCP*. Caso contrário seja (\bar{w}_r, \bar{z}_r) tal que

$$\bar{z}_r \bar{w}_r = \max\{\bar{z}_i \bar{w}_i : \bar{z}_i > 0, \bar{w}_i > 0\}.$$

Passo 4: *Ramificação*

Adicione dois novos nós $(k+1)$ e $(k+2)$ à lista \mathcal{L} , com programas quadráticos associados definidos por

$$QP(k+1) : I_{k+1} = I_t \cup \{r\}, J_{k+1} = J_t$$

$$QP(k+2) : I_{k+2} = I_t, J_{k+2} = J_t \cup \{r\}$$

e volte ao Passo 1.

Em [Júdice et al., 2002] é apresentada a implementação deste método em ambiente GAMS [Gill et al., 1981], um sistema desenvolvido em Fortran que incorpora um algoritmo de res-

trições activas MINOS para a determinação de um ponto estacionário de um programa quadrático. É evidente que este método enumerativo pode ser usado na prática para a resolução de *LCPs*. Notar que *LCPs* monótonos ou cuja matriz seja *RSuf* são resolvidos no primeiro nó, pois ou o *LCP* é não admissível ou então qualquer ponto estacionário do programa quadrático associado é solução do *LCP* (Teorema 1.2).

2.2 Algoritmo Complementar de Restrições Activas

Considere-se novamente o problema *MPEC* definido por

$$\begin{aligned}
 & \text{Minimizar} && f(w, z, y) \\
 & \text{sujeito a} && Ew = q + Mz + Ny \\
 & && Ay = b \\
 & && w \geq 0, z \geq 0, y \geq 0 \\
 & && z_i w_i = 0, \quad i = 1, \dots, n,
 \end{aligned} \tag{2.3}$$

onde $E \in \mathbb{R}^{l \times n}$, $M \in \mathbb{R}^{l \times n}$, $N \in \mathbb{R}^{l \times m}$, $y \in \mathbb{R}^m$, $z, w \in \mathbb{R}^n$, $q \in \mathbb{R}^l$, com $A \in \mathbb{R}^{p \times m}$ e $b \in \mathbb{R}^p$ e designe-se por \mathcal{K} o conjunto de restrições lineares que definem o *GLCP*. Começemos por introduzir a noção de ponto estacionário forte, devida a Scheel and Scholtes [Scheel e Scholtes, 2000], que é muitas vezes utilizada na literatura.

Definição 2.1 *Um ponto estacionário forte $\bar{x} = (\bar{w}, \bar{z}, \bar{y})$ do *MPEC* (2.3) é uma solução do *GLCP* que verifica condições de optimalidade de primeira ordem para o problema (2.3), ou seja, $\exists \lambda \in \mathbb{R}^l, \alpha, \beta, \delta \in \mathbb{R}^n, \nu \in \mathbb{R}^p, \gamma \in \mathbb{R}^m$:*

$$\begin{aligned}
 \nabla_w f(\bar{x}) &= E^T \lambda + \alpha + \bar{Z} \delta \\
 \nabla_z f(\bar{x}) &= -M^T \lambda + \beta + \bar{W} \delta \\
 \nabla_y f(\bar{x}) &= -N^T \lambda + A^T \nu + \gamma \\
 \alpha, \beta, \gamma &\geq 0 \\
 \alpha_i \bar{w}_i &= 0, & i = 1, \dots, n \\
 \beta_i \bar{z}_i &= 0, & i = 1, \dots, n \\
 \gamma_i \bar{y}_i &= 0, & i = 1, \dots, m,
 \end{aligned} \tag{2.4}$$

com \bar{W} e \bar{Z} matrizes diagonais de elementos diagonais \bar{w}_i e \bar{z}_i respectivamente.

O algoritmo Complementar de Restrições Activas introduzido por Scholtes em [Scholtes, 1999] foi melhorado e implementado em [Júdice et al., 2005]. A ideia base deste processo é utilizar em cada iteração soluções do *GLCP* que constitui as restrições do *MPEC*. Para isso o algoritmo começa por determinar uma solução do *GLCP*, o que pode ser feito usando o algoritmo enumerativo discutido anteriormente. Como as restrições complementares $z_i w_i = 0$

são mantidas em cada iteração, então um ponto estacionário do programa não linear

$$\begin{aligned}
 & \text{Minimizar} && f(w, z, y) \\
 & \text{sujeito a} && Ew = q + Mz + Ny \\
 & && Ay = b \\
 & && w \geq 0, z \geq 0, y \geq 0
 \end{aligned} \tag{2.5}$$

é um ponto estacionário forte, conforme mostra o seguinte teorema.

Teorema 2.1 *Se $(\bar{w}, \bar{z}, \bar{y})$ é um ponto estacionário do programa não linear (2.5) satisfazendo $z_i w_i = 0$ para todo $i = 1, \dots, n$, então $(\bar{w}, \bar{z}, \bar{y})$ é um ponto estacionário forte do MPEC (2.3).*

Demonstração: De acordo com a hipótese, $(\bar{w}, \bar{z}, \bar{y})$ satisfaz

$$\begin{aligned}
 \nabla_w f(\bar{w}, \bar{z}, \bar{y}) &= E^T \bar{\lambda} + \bar{\alpha} \\
 \nabla_z f(\bar{w}, \bar{z}, \bar{y}) &= -M^T \bar{\lambda} + \bar{\beta} \\
 \nabla_y f(\bar{w}, \bar{z}, \bar{y}) &= -N^T \bar{\lambda} + A^T \bar{\nu} + \bar{\gamma} \\
 \bar{\alpha}, \bar{\beta}, \bar{\gamma} &\geq 0 \\
 \bar{\alpha}_i \bar{w}_i &= 0, & i = 1, \dots, n \\
 \bar{\beta}_i \bar{z}_i &= 0, & i = 1, \dots, n \\
 \bar{\gamma}_i \bar{y}_i &= 0, & i = 1, \dots, m,
 \end{aligned}$$

com $\bar{\alpha}, \bar{\beta}, \bar{\gamma}$ multiplicadores de Lagrange associados às restrições de não negatividade e $\bar{\lambda}, \bar{\nu}$ multiplicadores associados às restrições de igualdade. Se $\bar{\delta} = 0$ então $(\bar{w}, \bar{z}, \bar{y}, \bar{\alpha}, \bar{\beta}, \bar{\gamma}, \bar{\lambda}, \bar{\nu}, \bar{\delta})$ satisfazem as condições (2.4) e portanto $(\bar{w}, \bar{z}, \bar{y})$ é um ponto estacionário forte do MPEC. \square

Tendo em conta este teorema, o algoritmo utiliza uma metodologia de restrições activas para o programa não linear (2.5), mantendo a complementaridade ao longo de todo o processo. Tal como é descrito em [Júdice et al., 2005], é apenas necessário modificar o critério para escolha da restrição a retirar do conjunto de restrições activas para um ponto não estacionário de (2.5) para obter um algoritmo que converge para um ponto estacionário forte do MPEC. O algoritmo é denominado de Complementar de Restrições Activas, pelo facto de apenas usar soluções complementares e empregar uma técnica de restrições activas. De forma a explicar os passos do algoritmo, em cada iteração k os iterandos (w, z, y) satisfazem as restrições do

problema (2.5) e o conjunto de restrições activas é constituído por:

$$\begin{aligned} Ew - Mz - Ny &= q \\ Ay &= b \\ w_i &= 0, i \in L_w \subseteq \{1, \dots, n\} \\ z_i &= 0, i \in L_z \subseteq \{1, \dots, n\} \\ y_i &= 0, i \in L_y \subseteq \{1, \dots, m\}, \end{aligned} \tag{2.6}$$

onde L_w, L_z e L_y são os conjuntos das restrições activas correspondentes às restrições de não negatividade nas variáveis w, z e y respectivamente. As restrições activas (2.6) podem sintetizar-se no seguinte sistema linear:

$$D_k x = g_k,$$

onde $x^T = [w, z, y]$ e $D_k \in \mathbb{R}^{t \times (2n+m)}$, com $t = l + p + |L_w| + |L_z| + |L_y|$ correspondendo o símbolo $|\cdot|$ à cardinalidade de conjunto.

As condições de optimalidade de primeira ordem para o problema de igualdades

$$\begin{aligned} & \text{Minimizar} \quad f(x) \\ & \text{sujeito a} \quad D_k x = g_k \end{aligned} \tag{2.7}$$

podem ser escritas na forma

$$\begin{aligned} \nabla f(x) &= D_k^T \mu \\ D_k x &= g_k. \end{aligned}$$

Assumindo a hipótese de não degenerescência

$$t \leq 2n + m, \quad c(D_k) = t$$

onde $c(D_k)$ representa a característica da matriz D_k , é possível calcular univocamente o vector dos Multiplicadores de Lagrange μ , que é decomposto nos seguintes subvectores:

β : subvector associado ao primeiro conjunto de restrições em (2.6).

ϑ : subvector associado ao segundo conjunto de restrições em (2.6).

λ_i^x : subvector associado a $x_i = 0$ nos três últimos conjuntos de restrições de (2.6).

O algoritmo pode ser sintetizado nos seguintes passos:

ALGORITMO COMPLEMENTAR DE RESTRIÇÕES ACTIVAS

Passo 0: *Ponto Inicial*

Faça $k = 1$, resolva o *GLCP* associado ao *MPEC* e seja x^k a solução obtida. Seja $D_k x = g_k$

o conjunto das restrições activas em x^k e L_w, L_z e L_y os conjuntos das restrições activas correspondentes às restrições de não negatividade nas variáveis w, z e y , respectivamente.

Passo 1: *Condições de Optimalidade*

1. Se x^k é um ponto estacionário do Problema de Igualdades (2.7), isto é, se existe um único μ , no sentido dos mínimos quadrados, tal que

$$D_k^T \mu = \nabla f(x^k) \Leftrightarrow D_k D_k^T \mu = D_k \nabla f(x^k)$$

então:

a) Se

$$\lambda_i^y \geq 0 \quad \forall i \in L_y$$

$$\lambda_i^z \geq 0 \quad \forall i \in L_z \cap L_w$$

$$\lambda_i^w \geq 0 \quad \forall i \in L_z \cap L_w$$

pare: x^k é ponto estacionário do *MPEC*.

b) Se existe pelo menos um i tal que

$$\lambda_i^y < 0 \quad \text{para } i \in L_y \quad \text{ou}$$

$$\lambda_i^z < 0 \quad \text{para } i \in L_z \cap L_w \quad \text{ou}$$

$$\lambda_i^w < 0 \quad \text{para } i \in L_z \cap L_w$$

retire do conjunto de restrições activas corrente uma restrição $y_i = 0$, ou $z_i = 0$ ou $w_i = 0$, escolhendo por exemplo, a que corresponde ao multiplicador de Lagrange mais negativo. Seja $D_{k_i} x = g_{k_i}$ a linha que é retirada do conjunto de restrições activas, e reordene as linhas da seguinte forma

$$D_k = \begin{bmatrix} \bar{D}_k \\ D_{k_i} \end{bmatrix}, \quad g_k = \begin{bmatrix} \bar{g}_k \\ g_{k_i} \end{bmatrix}.$$

Encontre uma direcção d tal que $\nabla f(x^k)^T d < 0$, $\bar{D}_k d = 0$, e $D_{k_i} d > 0$. Substitua D_k por \bar{D}_k e vá para o Passo 3.

2. Se a solução x^k não é um ponto estacionário do Problema de Igualdades (2.7), vá para o Passo 2.

Passo 2: *Determinação da Direcção de Pesquisa*

Determine uma direcção descendente para f no conjunto das restrições activas, isto é, encontre d tal que:

$$\nabla f(x^k)^T d < 0$$

$$D_k d = 0.$$

Passo 3: *Determinação do Passo*

1. Calcule o valor máximo α_{max} de α tal que

$$x^k + \alpha d \geq 0.$$

Para isso basta determinar α_{max} pelo seguinte processo

$$\min \left\{ \frac{x_i^k}{-d_i^k} : d_i^k < 0, \text{ onde } i \text{ está associado às restrições inactivas } x_i \geq 0 \right\}.$$

2. Determine $0 < \alpha_k \leq \alpha_{max}$ tal que

$$x^k + \alpha_k d$$

permita um decréscimo significativo em f . Para tal utilize uma técnica de pesquisa em linha. Se $\alpha_k = +\infty$, pare: o *MPEC* é ilimitado.

Passo 4: *Actualização*

Faça

$$x^{k+1} = x^k + \alpha_k d.$$

Se $\alpha_k = \alpha_{max}$, adicione ao conjunto de restrições activas as restrições $x_i \geq 0$ onde α_{max} foi atingido, por forma a que o pressuposto da não degenerescência seja mantido. Caso contrário mantenha o conjunto das restrições activas. Volte ao Passo 1.

Este algoritmo foi desenvolvido em [Júdice et al., 2005] e implementado no ambiente MINOS. Nessa implementação, em cada iteração do método de restrições activas, o sistema de restrições é decomposto na forma

$$Bx_B + Sx_S + Nx_N = 0,$$

onde B é uma matriz base quadrada não singular e x_B , x_S , x_N são os vectores das variáveis básicas, superbásicas e não básicas, respectivamente. Em qualquer solução usada pelo processo, as variáveis básicas e superbásicas tomam valores entre os seus limites inferior e superior (podem ser iguais a um dos limites), enquanto que as variáveis não básicas tomam valores iguais a um dos seus limites inferior ou superior.

Sempre que o multiplicador de lagrange associado a uma restrição activa $x_r = l_r$ (normalmente igual a zero) ou $x_r = u_r$ não tem o sinal adequado, a correspondente variável não básica x_r é seleccionada para mudar de estatuto, passando a superbásica ou básica. Consequentemente, se uma variável básica ou superbásica atinge um dos seus limites, a variável passa a não básica e neste último caso, a cardinalidade de S é diminuída em uma unidade. Se a variável não tiver complementar, então pode mudar de valor sem acrescentar qualquer

difficuldade ao método. O mesmo acontece se a sua complementar é também uma variável não básica de valor nulo. Se nenhuma destas duas situações acontecer, isto é, se a complementar da variável a escolher é básica, superbásica ou não básica com valor positivo, então a variável candidata não pode mudar o seu estatuto e uma nova variável não básica deve ser seleccionada. O algoritmo termina quando nenhuma das variáveis não básicas candidatas a mudar de estatuto o pode fazer. Como é provado em [Júdice et al., 2005] o algoritmo complementar de restrições activas possui convergência global para um ponto estacionário forte sob uma hipótese de não degenerescência. Além disso, no mesmo artigo são discutidos alguns casos em que a convergência do processo é finita. Experiência computacional descrita em [Júdice et al., 2005] indica a eficiência do algoritmo para a determinação de pontos estacionários fortes. O algoritmo mostra-se competitivo e em geral mesmo superior a outras técnicas de *penalty* [Facchinei et al., 1999] e de pontos interiores [Fukushima et al., 1998] para a resolução de *MPECs* com restrições lineares. Por essa razão não iremos abordar essas técnicas alternativas neste trabalho.

2.3 Algoritmo Sequencial Complementar

Este método foi inicialmente desenvolvido por Júdice e Faustino [Júdice e Faustino, 1992] para determinar a solução óptima de *MPECs* lineares com restrições lineares. Para a sua descrição consideremos o *MPEC* linear

$$\begin{aligned} \text{Minimizar} \quad & c^T z + d^T y = f(z, y) \\ \text{sujeito a} \quad & w = q + Mz + Ny \\ & Ay = b \\ & w \geq 0, z \geq 0, y \geq 0 \\ & z^T w = 0, \end{aligned}$$

onde $z, w, q, c \in \mathbb{R}^n$, $d \in \mathbb{R}^m$, $M \in \mathbb{R}^{n \times n}$, $N \in \mathbb{R}^{n \times m}$, $y \in \mathbb{R}^m$, $b \in \mathbb{R}^p$ e $A \in \mathbb{R}^{p \times m}$, com $p < m$.

O processo assume a existência de uma restrição linear $c^T z + d^T y \leq \lambda_k$, no conjunto das restrições do problema, com λ_k um parâmetro que impede a função $c^T z + d^T y$ de obter um valor superior ou igual ao valor de uma outra solução do *GLCP* obtida anteriormente. Por introdução dessa restrição adicional obtém-se um problema linear complementar generalizado da forma

$$\begin{aligned} \text{GLCP}(\lambda_k) \quad & \begin{bmatrix} w \\ 0 \\ \mu \end{bmatrix} = \begin{bmatrix} q \\ b \\ \lambda_k \end{bmatrix} + \begin{bmatrix} M & N \\ 0 & -A \\ -c^T & -d^T \end{bmatrix} \begin{bmatrix} z \\ y \end{bmatrix} \\ & w, z, y, \mu \geq 0 \\ & w^T z = 0. \end{aligned} \tag{2.8}$$

O processo consiste em resolver uma sucessão de $GLCP(\lambda_k)$, com $\{\lambda_k\}$ uma sucessão estritamente decrescente. Assim o processo começa por resolver o $GLCP(\lambda_0)$ obtido do problema (2.8) por omissão da restrição $c^T z + d^T y \leq \lambda_k$. Em cada iteração $k \geq 1$, o parâmetro λ_k é actualizado por

$$\lambda_k = f(z^{k-1}, y^{k-1}) - \gamma |f(z^{k-1}, y^{k-1})|, \quad (2.9)$$

em que (z^{k-1}, y^{k-1}) é a solução do $GLCP(\lambda_{k-1})$ e γ é um parâmetro positivo pequeno, previamente estabelecido.

Se para um determinado valor do parâmetro o $GLCP(\lambda_k)$ encontra uma solução, então o algoritmo complementar de restrições activas é accionado para obter um ponto estacionário do $MPEC$. Se esse $GLCP$ não tem solução, então o último ponto estacionário obtido é uma solução global óptima aproximada, que para uma boa actualização do parâmetro λ_k , fornece um mínimo global para o $MPEC$. Os passos do algoritmo apresentam-se seguidamente.

ALGORITMO SEQUENCIAL COMPLEMENTAR

Passo 0: Faça $k = 0$.

Passo 1: Resolva o $GLCP(\lambda_k)$. Se o problema não tem solução vá para Passo 3. Caso contrário seja $\bar{x} = (\bar{z}, \bar{y})$ a solução do $GLCP(\lambda_k)$. Determine um ponto estacionário $x^k = (z^k, y^k)$ para o $MPEC$, utilizando o método complementar de restrições activas com \bar{x} o ponto inicial.

Passo 2: Actualize λ_k de acordo com (2.9). Faça $k \leftarrow k + 1$ e vá para o Passo 1.

Passo 3: Se $k = 0$, o $MPEC$ não tem solução. Caso contrário $x^{k-1} = (z^{k-1}, y^{k-1})$ é uma solução ε -óptima do $MPEC$, com ε dado por $\varepsilon = \gamma |f(z^{k-1}, y^{k-1})|$.

O algoritmo é assim um processo inteligente de pesquisar pontos estacionários do $MPEC$ até à obtenção de um que seja um mínimo global. A introdução da restrição linear nas restrições do $GLCP(\lambda_k)$ torna-o num problema NP-difícil, mesmo no caso do $GLCP$ inicial ser um problema simples, como acontece quando a matriz M é PSD [Júdice e Vicente, 1994]. Assim todos os $GLCPs$, com a possível excepção do primeiro, necessitam de uma técnica enumerativa para a sua resolução. A implementação do processo combina as duas técnicas referidas anteriormente, com a inclusão de algumas heurísticas que melhoram o desempenho do método. Um desses processos consiste em utilizar a solução básica do $GLCP(\lambda_{k-1})$ como solução inicial na resolução do $GLCP(\lambda_k)$. Apesar de esta solução não ser uma solução admissível para o $GLCP(\lambda_k)$, conduz normalmente a uma grande redução no esforço computacional do método enumerativo, uma vez que os dois problemas apenas diferem na última componente do vector dos termos independentes.

De acordo com a exposição do algoritmo, o último $GLCP$ não pode ter solução para que o

algoritmo conclua que o último ponto estacionário do *MPEC* obtido é o seu mínimo global. Como referimos anteriormente, o método enumerativo tem grande dificuldade em terminar neste caso. Experiência computacional com *MPECs* de diferentes origens apresentada em [Júdice e Faustino, 1991, Júdice e Faustino, 1992] indica que o algoritmo sequencial é em geral bastante eficaz para obter um mínimo global, mas encontra muitas dificuldades em provar que essa solução óptima foi realmente obtida. Esse tipo de comportamento será confirmado nas experiências por nós efectuadas e que serão relatadas nesta tese.

O algoritmo sequencial complementar pode ser estendido para a resolução de um *MPEC* com um função quadrática convexa. Neste caso o método enumerativo deve ser modificado de modo a poder resolver problemas de optimização com restrições lineares e uma restrição quadrática convexa. Um algoritmo de restrições activas ou de pontos interiores terá de ser usado em cada nó do método enumerativo para a determinação de um ponto estacionário da função quadrática de complementaridade num conjunto admissível constituído por restrições lineares e uma restrição quadrática convexa.

2.4 Algoritmo Complementar de Pesquisa em Árvore

Considere-se novamente o *MPEC*

$$\begin{aligned} & \text{Minimizar} && f(z, y) \\ & \text{sujeito a} && Ew = q + Mz + Ny \\ & && Ay = b \\ & && w \geq 0, z \geq 0, y \geq 0 \\ & && z^T w = 0, \end{aligned}$$

onde $E \in \mathbb{R}^{l \times n}$, $M \in \mathbb{R}^{l \times n}$, $N \in \mathbb{R}^{l \times m}$, $y \in \mathbb{R}^m$, $z, w \in \mathbb{R}^n$, $q \in \mathbb{R}^l$, com $A \in \mathbb{R}^{p \times m}$ e $b \in \mathbb{R}^p$, onde assumimos que f é uma função convexa no conjunto das restrições lineares do problema.

Tal como nos problemas inteiros binários, este método utiliza o processo clássico de ramificação em árvore gerada pela dicotomia das variáveis complementares já referida anteriormente. Além disso, o algoritmo assume que os *GLCPs* associados aos vários nós da árvore se podem resolver sem recurso a um método enumerativo.

Como algoritmo de ramificação e limitação, o processo vai calculando limites inferiores e superiores à medida que a ramificação é efectuada, por forma a diminuir o intervalo de pesquisa e acelerar a convergência do processo. Após a determinação de uma solução do *GLCP* associada a um nó, o método complementar de restrições activas é empregue para obter um ponto estacionário do *MPEC* e assim permitir a determinação de um possível novo limite superior para a função objectivo. O limite inferior é determinado através da

relaxação convexa do problema, obtida por omissão da restrição de complementaridade.

A escolha do par de variáveis a ramificar em cada nó é feita a partir dos custos reduzidos da função objectivo referentes aos pontos estacionários associados aos diferentes nós. Tal como é discutido em [Júdice et al., 2006b], há a possibilidade de usar uma ramificação dicotómica simples ou uma ramificação mais profunda. Nesta tese apenas consideraremos a primeira hipótese. Como é usual neste tipo de técnicas de pesquisa em árvore com limites, é ainda possível adicionar técnicas de cortes de modo a melhorar os limites inferiores e assim reduzir a pesquisa na árvore para um mínimo global do *MPEC*. Cortes disjunctivos foram sugeridos em [Júdice et al., 2006b] sem muito sucesso, pelo que não serão considerados na descrição do algoritmo e nas nossas experiências.

Na explicação dos passos do algoritmo, usaremos as notações

$$I_k = \{i : z_i = 0 \text{ fixa}\} \quad \text{e} \quad J_k = \{i : w_i = 0 \text{ fixa}\}$$

para representar os conjuntos de índices correspondentes às variáveis fixas em zero ao longo do caminho da árvore percorrido até ao nó k . Além disso $\text{VAL}(k)$ é o valor do limite superior associado ao nó k e \mathcal{L} a lista de nós por explorar.

ALGORITMO COMPLEMENTAR DE PESQUISA EM ÁRVORE

Passo 0: *Inicialização*

$I_1 = J_1 = \emptyset$, $\mathcal{L} = \{1\}$, $\text{UB} = +\infty$ e $\text{VAL}(1) = +\infty$.

Passo 1: *Seleção do subproblema*

Se $\mathcal{L} = \emptyset$ e $\text{UB} = +\infty$, então o *MPEC* não tem solução óptima (o *GLCP* não tem solução ou o *MPEC* é ilimitado). Se $\mathcal{L} = \emptyset$ e $\text{UB} < +\infty$ a solução $(\bar{z}, \bar{w}, \bar{y})$ associada a UB é o mínimo global do *MPEC* e o valor óptimo é UB . De outro modo escolha um nó k na lista \mathcal{L} associado ao menor valor $\text{VAL}(k)$ com conjuntos I_k e J_k de variáveis z_i e w_i fixas em zero e solução inicial $x^k = (z^k, w^k, y^k)$ do *GLCP* associado. Faça $\mathcal{L} = \mathcal{L} \setminus \{k\}$.

Passo 2: *Determinação de um Limite Superior*

Aplique um algoritmo de restrições activas para encontrar um ponto estacionário do programa quadrático

$$\begin{aligned} QP(k) : \quad & \text{Minimizar} \quad z^T w \\ & \text{sujeito a} \quad w = q + Mz + Ny \\ & \quad \quad \quad Ay = b \\ & \quad \quad \quad z \geq 0, w \geq 0, y \geq 0 \\ & \quad \quad \quad z_i = 0, \quad i \in I_k \\ & \quad \quad \quad w_i = 0, \quad i \in J_k \end{aligned}$$

começando com a solução x^k associada a esse nó. Se o problema é não admissível vá para o Passo 1. De outro modo determine um ponto estacionário $x^k = (z^k, w^k, y^k)$ do *MPEC*,

cujo $GLCP$ é induzido por este programa quadrático, utilizando o algoritmo complementar de restrições activas. Atualize UB

$$UB = \min \left\{ UB, f(z^k, y^k) \right\},$$

armazenando a solução incumbente se UB diminuiu de valor.

Passo 3: *Determinação de um Limite Inferior*

Determine um limite inferior, $LB(k)$ para o nó k , usando como solução inicial x^k . Se $LB(k) \geq UB$ vá para o Passo 1.

Passo 4: *Ramificação*

Identifique os índices das variáveis w e z cujos multiplicadores de Lagrange associados às restrições $z_i \geq 0$ e $w_j \geq 0$ no ponto estacionário x^k são negativos para $i \notin I_k$ e $j \notin J_k$. Seja r o índice da variável que apresenta o menor multiplicador. Adicione à lista \mathcal{L} dois subproblemas com conjuntos, respectivamente

$$I_{|\mathcal{L}|+1} = I_k \cup \{z_r\} \text{ e } J_{|\mathcal{L}|+1} = J_k$$

$$I_{|\mathcal{L}|+2} = I_k \text{ e } J_{|\mathcal{L}|+2} = J_k \cup \{w_r\}.$$

Para cada um desses nós considere a solução inicial $x^k = (z^k, w^k, y^k)$ e faça $VAL(k) = f(z^k, y^k)$. Volte ao Passo 1.

A implementação deste algoritmo pode também ser feita em ambiente MINOS, sendo para isso necessário que o $GLCP$ de cada um dos nós se possa resolver a partir de um ponto estacionário da função complementaridade no conjunto convexo definido pelas restrições lineares associadas a esse nó. Tal como é discutido em [Júdice et al., 2006b], este tipo de implementação pode ser utilizado para $GLCPs$ da forma (1.4) discutidos no capítulo 1. A experiência computacional revela que o processo é bastante eficiente na determinação da solução óptima de $MPECs$ com valor óptimo conhecido. Se esta hipótese não for verificada, a eficácia do algoritmo diminui com o aumento da dimensão do problema.

2.5 Algoritmo Enumerativo Paramétrico

Considere-se o $MPEC$ linear com restrições lineares definido por

$$\begin{aligned} & \text{Minimizar} && c^T z + d^T y \\ & \text{sujeito a} && Ew = q + Mz + Ny \\ & && Ay = b \\ & && w \geq 0, z \geq 0, y \geq 0 \\ & && z^T w = 0, \end{aligned} \tag{2.10}$$

onde $E \in \mathbb{R}^{l \times n}$, $M \in \mathbb{R}^{l \times n}$, $N \in \mathbb{R}^{l \times m}$, $y \in \mathbb{R}^m$, $z, w \in \mathbb{R}^n$, $q \in \mathbb{R}^l$, $b \in \mathbb{R}^p$ e $A \in \mathbb{R}^{p \times m}$ com $p < m$.

Em [Bard e Moore, 1990] é discutido um algoritmo de ramificação e limitação para a determinação do mínimo global do problema (2.10). Em cada nó da árvore é determinado um limite inferior para o *MPEC* a partir da solução óptima do programa linear obtido do *MPEC* por relaxação das restrições complementares. Sempre que a solução óptima satisfazer a condição de complementaridade, o limite superior é actualizado e o processo prossegue com um outro nó. O algoritmo termina quando não houver nenhum nó aberto cujo limite inferior seja menor do que o menor limite superior obtido durante a pesquisa. Experiência computacional descrita em [Júdice e Faustino, 1992, Júdice e Faustino, 1991] indicou que o método não é muito eficiente para o fim em vista.

Para melhorar a eficiência do algoritmo sugerimos a utilização em cada nó de um algoritmo de optimização local para a determinação de um ponto estacionário do programa quadrático que minimiza a função complementaridade sujeita às restrições lineares nesse nó. Além disso, sempre que uma solução de um *GLCP* é obtida, o algoritmo complementar de restrições activas é empregue para obter um ponto estacionário do *MPEC*. De modo a evitar a repetição de pontos estacionários e, à semelhança do algoritmo sequencial complementar, é usado um corte que força a função objectivo a reduzir estritamente de valor à medida que se vão gerando os sucessivos pontos estacionários. É importante notar que o algoritmo pressupõe que o programa linear relaxado associado ao *MPEC* tem solução óptima. Os passos do algoritmo são apresentados a seguir.

ALGORITMO ENUMERATIVO PARAMÉTRICO

Passo 0: *Inicialização*

Seja λ_1 um número positivo grande e $\mathcal{L} = \{1\}$ a lista de nós em aberto. Determine a solução óptima do Programa Linear (*LP*) relaxado (z^*, y^*) e faça $\text{VAL}(1) = c^T z^* + d^T y^*$.

Passo 1: *Avaliação da optimalidade*

Se $\mathcal{L} = \emptyset$, então o último ponto estacionário obtido é a solução óptima do *MPEC* e o valor óptimo é UB.

Passo 2: *Escolha do nó a ramificar*

Determine o nó $t \in \mathcal{L}$, tal que

$$\text{VAL}(t) = \min\{\text{VAL}(i) : i \in \mathcal{L}\}.$$

Passo 3: *Determinação de Ponto estacionário do GLCP*

Determine o ponto estacionário (\bar{z}, \bar{y}) de

$$\begin{aligned} & \text{Minimizar} && z^T w \\ & \text{sujeito a} && w = q + Mz + Ny \\ & && z \geq 0, w \geq 0, y \in K_y \\ & && z_i = 0, i \in I \\ & && w_j = 0, j \in J \\ & && c^T z + d^T y \leq \lambda_k, \end{aligned}$$

com I e J os conjuntos dos índices das variáveis marcadas nesse nó. Para esse efeito utilize o método de restrições activas com ponto inicial, a solução associada a $\text{VAL}(t)$.

Passo 4: *Determinação de Ponto estacionário do MPEC*

Se o ponto estacionário (\bar{z}, \bar{y}) é uma solução complementar, aplique o método complementar de restrições activas com ponto inicial (\bar{z}, \bar{y}) e determine um ponto estacionário (\tilde{z}, \tilde{y}) do *MPEC*. Actualize

$$\text{UB} = c^T \tilde{z} + d^T \tilde{y}$$

$$\lambda_k = \text{UB} - \gamma |\text{UB}|$$

$$\mathcal{L} = \mathcal{L} \setminus \{t \in \mathcal{L} : \text{VAL}(t) \geq \text{UB}\},$$

com $\gamma > 0$ e pequeno e volte ao Passo 1.

Passo 5: *Escolha da variável de ramificação*

Se o ponto estacionário (\bar{z}, \bar{y}) não é uma solução complementar, escolha s a partir de

$$\bar{z}_s \bar{w}_s = \max\{\bar{z}_i \bar{w}_i : \bar{z}_i > 0, \bar{w}_i > 0\}.$$

Passo 6: *Determinação do número de ramos*

Considere os nós associados a $z_s = 0$ e $w_s = 0$ e resolva os *LPs* associados

$$\begin{aligned} & \text{Minimizar} && c^T z + d^T y \\ & \text{sujeito a} && w = q + Mz + Ny \\ & && z \geq 0, w \geq 0, y \in K_y \\ & && z_i = 0, i \in I \\ & && w_j = 0, j \in J, \end{aligned}$$

com $I = I \cup \{s\}$ e $J = J \cup \{s\}$ respectivamente. Para qualquer um desses *LPs* admissíveis faça

$$\text{VAL}(|\mathcal{L}| + r) = c^T z^* + d^T y^*,$$

com $r = 1, 2$ e (z^*, y^*) a solução óptima do *LP* respectivo. Volte ao Passo 1, com

$$\mathcal{L} = \mathcal{L} \setminus \{s\} \cup \{|\mathcal{L}| + r : r \in \{0, 1, 2\}\},$$

dependendo do número de LPs admissíveis.

Com esta metodologia, o algoritmo enumerativo paramétrico vai percorrer o mesmo caminho desde o nó original até aos nós em aberto, associando deste modo uma obrigação de escolha de determinada pesquisa na resolução de todos os $GLCPs$. Poderá acontecer que um nó gerado antes da actualização do parâmetro λ_k não venha a poder ser gerado em fase posterior. Essa situação é identificada pela existência de um nó descendente t por explorar, com valor $VAL(t)$ superior ao valor de λ_k actual.

Contrariamente ao algoritmo complementar da secção 2.4, este método de pesquisa em árvore pode ser usado para quaisquer $GLCPs$, uma vez que não é obrigatória a resolução do $GLCP$ em cada um dos nós visitados durante o processo. Além disso a sua implementação pode ser efectuada usando o ambiente MINOS, pois apenas necessita do cálculo de soluções óptimas de programas lineares e de pontos estacionários de programas quadráticos. Como última vantagem deste método refira-se que, à semelhança do algoritmo sequencial complementar, o processo pode ser estendido a $MPECs$ com funções quadráticas convexas, de acordo com as observações relatadas na secção 2.3.

2.6 Experiência Computacional

Nesta secção faz-se uma comparação do desempenho dos diversos algoritmos descritos anteriormente, através da resolução de dois tipos de problemas de teste que se descrevem seguidamente.

(I) Descrição dos problemas

Problemas Tipo I. Problemas de dois níveis lineares

Os problemas de dois níveis lineares utilizados, designados por BL./BL.M, foram construídos recorrendo ao gerador apresentado em [Calamai e Vicente, 1993]. Estes problemas são transformados em problemas $MPEC$, apresentando a vantagem de ser conhecida a sua solução óptima, assim como o número total de mínimos locais e globais.

As características dos problemas usados na experiência computacional apresentam-se na Tabela 2.1, sendo nx e ny o número de variáveis do primeiro e segundo nível respectivamente no gerador referido, M e N o número de linhas e colunas respectivamente do problema $MPEC$ resultante e OBJ o valor do mínimo global do problema.

A designação de BL._M refere-se a problemas com as mesmas características dos problemas BL. deste gerador, mas em que a matriz do problema de segundo nível é mais densa, devido

a uma transformação no problema referida nesse artigo.

Tabela 2.1: *Características dos problemas de dois níveis.*

PROBLEMA	nx	ny	M	N	OBJ
BL ₁	4	2	12	28	-6
BL ₂	10	5	30	70	-14
BL ₃	15	10	55	125	-14.8
BL ₄	10	15	60	130	8
BL ₅	30	20	110	250	-66.5
BL ₆	40	40	200	440	-51
BL ₇	50	60	270	590	-60

Problemas Tipo II. *Problemas Saco-Mochila*

Os problemas de saco-mochila consistem em determinar um vector $x \in \mathbb{R}^n$ tal que

$$\begin{aligned} a^T x &= b \\ x_i &\in \{0, 1\}, \quad i = 1, \dots, n \end{aligned}$$

com $b \in \mathbb{R}^+$ e $a \in \mathbb{R}^{+n}$. Tal como foi descrito em [Júdice et al., 2002], as componentes a_i do vector positivo $a \in \mathbb{R}^n$ foram geradas aleatoriamente com distribuição uniforme no intervalo $[1, 50]$ e o termo independente b determinado pela equação

$$b = \sum_{i \in I} a_i,$$

sendo I um subconjunto de $\{1, \dots, n\}$ que representa a percentagem de variáveis x_i com valor não nulo, numa solução do problema. A transformação do problema saco-mochila num problema linear complementar foi efectuada de três formas distintas que se expõem seguidamente.

- 1) A primeira formulação baseia-se no facto das restrições

$$x_i \in \{0, 1\}, \quad i = 1, \dots, n$$

se poderem escrever na forma equivalente

$$w_i = 1 - x_i, \quad x_i \geq 0, \quad w_i \geq 0, \quad x_i w_i = 0, \quad i = 1, \dots, n.$$

Como é provado em [Chung, 1989] o problema saco-mochila é então equivalente ao *LCP* de dimensão $(n + 2)$

$$\begin{aligned} w &= e - x \\ w_{n+1} &= -b + a^T x - \alpha x_{n+1} \\ w_{n+2} &= b - a^T x - \beta x_{n+2} \\ x_i &\geq 0, \quad w_i \geq 0, \quad x_i w_i = 0, \quad i = 1, \dots, n + 2, \end{aligned}$$

com $e = [1 \ 1 \ \dots \ 1]^T \in \mathbb{R}^n$ e $\alpha, \beta \in \mathbb{R}^+$. A matriz M do LCP assume o seguinte aspecto:

$$M = \begin{bmatrix} -I & 0 & 0 \\ a^T & -\alpha & 0 \\ -a^T & 0 & -\beta \end{bmatrix} \in \mathbb{R}^{(n+2) \times (n+2)}$$

sendo I a matriz identidade de ordem n . É fácil de ver que a matriz M poderá ser NSD ou IND dependendo da escolha das constantes α e β . Com efeito tem-se

- $M \in NSD$, se α e β forem escolhidos de forma a satisfazerem as condições

$$\alpha > \theta \frac{a^T a}{4} \quad \text{e} \quad \beta > \theta \alpha \frac{a^T a}{4\alpha - a^T a}, \quad \text{com } \theta > 1 \text{ arbitrário.}$$

- $M \in IND$, se $\alpha > \theta \frac{a^T a}{4}$ e $\beta < \frac{\alpha}{\theta} \frac{a^T a}{4\alpha - a^T a}$, com $\theta > 1$.

- 2) O problema é também equivalente a um problema linear complementar de dimensão $(n+2)$ atendendo ao resultado estabelecido em [Murty e Júdice, 1996]. Nesse caso o LCP tem o vector q e a matriz M seguintes:

$$q = [a \ -b \ b]^T \in \mathbb{R}^{n+2},$$

$$M = \begin{bmatrix} -I & e & -e \\ e^T & -2n & 0 \\ -e^T & 0 & -2n \end{bmatrix} \in \mathbb{R}^{(n+2) \times (n+2)},$$

com I a matriz identidade de ordem n e $e = [1 \ 1 \ \dots \ 1]^T \in \mathbb{R}^n$. Note-se que neste caso a matriz M é simétrica NSD .

- 3) Uma terceira formulação do problema pode ser encontrada em [Kojima et al., 1991] e obtém um LCP , em que q e M são dados por

$$q = [p \ p \ \dots \ p \ -b \ b]^T \in \mathbb{R}^{4n+2},$$

$$M = \begin{bmatrix} B & 0 & \dots & 0 & 0 & 0 \\ 0 & B & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & B & 0 & 0 \\ & & \bar{a}^T & & & \\ & & -\bar{a}^T & & & \end{bmatrix} \in \mathbb{R}^{(4n+2) \times (4n+2)},$$

com $p = [0 \ 0 \ -1 \ 1]^T \in \mathbb{R}^4$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4}$$

e o vector $\bar{a} \in \mathbb{R}^{4n+2}$ tem componentes

$$\bar{a}_i = \begin{cases} a_j, & \text{se } i = 4j - 3 \text{ com } j = 1, \dots, n \\ 0, & \text{caso contrário.} \end{cases}$$

Neste caso a matriz M é uma matriz P_0 .

Estas três formulações do Problema Saco-Mochila permitiram construir um conjunto de *LCPs* NP-difíceis. Na tabela com os resultados computacionais usamos as letras N e I para distinguir na primeira formulação a geração de matrizes NSD e IND respectivamente. Na denominação do problema são ainda incorporados uma letra e um algarismo. As letras A, B , ou C dizem respeito à percentagem de variáveis não nulas numa solução do problema ser de 75%, 50% ou 25% respectivamente, enquanto que a designação numérica do problema refere-se à dimensão do mesmo, sendo utilizados os dígitos 1, 2, 3 e 4 relativos à dimensão 22, 52, 102 e 152 nas duas primeiras formulações, sendo que na terceira formulação os dígitos 2 e 4 referem-se a dimensões 50 e 150 respectivamente.

(II) Resultados Computacionais

Nas tabelas, a apresentar nesta secção, NIS e NOS representam respectivamente o número de iterações (operações pivotais) e nodos utilizados até encontrar a solução óptima do *MPEC* e NOT e NIT são respectivamente o número total de nodos e de iterações gastas pelo método. Efectuou-se inicialmente uma experiência computacional utilizando problemas de dois níveis na avaliação do desempenho dos algoritmos descritos anteriormente para a resolução de *MPECs*. A análise de resultados incluídos na Tabela 2.2 indica que o algoritmo complementar de pesquisa em árvore é capaz de resolver eficientemente os problemas de pequena dimensão, requerendo no entanto um grande esforço para verificar a sua optimalidade. Por sua vez, o algoritmo sequencial complementar apresenta um desempenho bastante bom na determinação da solução óptima destes problemas, nomeadamente no número de iterações usadas até à sua obtenção. Quando se compara o esforço computacional para a obtenção da solução óptima com o esforço total do processo, verifica-se que a diferença não é muito significativa na maioria dos testes, pelo que o método sequencial complementar é mais robusto do que o algoritmo complementar de pesquisa em árvore. Atenda-se a que nos dois últimos exemplos da Tabela 2.2 o método complementar de pesquisa em árvore não foi capaz

Tabela 2.2: *Resolução dos problemas de dois níveis lineares*

	PROBLEMA	SEQUENCIAL COMPLEMENTAR				ENUMERATIVO PARAMÉTRICO				COMPLEMENTAR DE PESQUISA EM ÁRVORE			
		NIS	NOS	NIT	NOT	NIS	NOS	NIT	NOT	NIS	NOS	NIT	NOT
Dois Níveis	BL ₁	8	1	19	20	26	4	43	15	21	2	27	9
	BL _{1M}	10	1	33	35	66	13	80	22	10	1	37	5
	BL ₂	34	10	149	167	149	17	324	96	120	16	255	111
	BL _{2M}	21	1	120	155	101	8	296	84	21	1	219	67
	BL ₃	57	14	2480	2520	4362	604	5748	1612	484	109	10270	6125
	BL _{3M}	58	15	2322	2452	2927	324	5088	1185	206	5	5505	1506
	BL ₄	96	52	3785	4367	689	46	14778	4206	2164	614	6086	1695
	BL ₅	99	25	682	641	271	20	1578	488	248	29	1140	575
	BL ₆	255	94	50255 ⁽¹⁾	41598	234711 ⁽¹⁾	13889	266202	15935	14003 ⁽³⁾	7584	50001	31817
	BL ₇	1721	1245	51723 ⁽¹⁾	46798	1616 ⁽²⁾	886	259670	7773	49357 ⁽³⁾	29999	50013	30474

⁽¹⁾ Foi atingido o número máximo de iterações permitidas no método enumerativo (50000).

⁽²⁾ O valor obtido foi de -48, pelo que a solução não é ótima.

⁽³⁾ A solução ótima não foi obtida porque o limite máximo de iterações impostas (50000) foi excedido. Os valores obtidos foram respectivamente -49 e -41.

de obter a solução óptima do *MPEC* com a utilização de um número bastante elevado de iterações.

Tabela 2.3: *Resolução dos LCPs associados a problemas saco-mochila*

	PROBLEMA	COMPLEMENTAR DE				SECUENCIAL	
		ENUMERATIVO		PESQUISA EM ÁRVORE		COMPLEMENTAR	
		NIT	NOT	NIT	NOT	NIT	NOT
SACO-MOCHILA - 1º MÉTODO	K _{NA1}	171	31	66	2	312	115
	K _{NA2}	620	90	11894	280	727	243
	K _{NA3}	147	20	206	2	1649	426
	K _{NA4}	228	31	528	2	114	6
	K _{NB1}	352	91	304	18	2596	1337
	K _{NB2}	60	9	72	2	4420	1650
	K _{NB3}	207	33	415	2	782	184
	K _{NB4}	200	32	205	2	1295	311
	K _{NC1}	226	55	163	6	156	19
	K _{NC2}	47	8	217	2	1198	310
	K _{NC3}	195	30	208	2	536	55
	K _{NC4}	131	22	162	2	755	181
	K _{IA1}	171	31	66	2	312	115
	K _{IA2}	620	90	11886	280	749	243
	K _{IA3}	147	20	206	2	1651	426
	K _{IA4}	228	31	528	2	114	6
	K _{IB1}	352	91	304	18	2708	1337
	K _{IB2}	60	9	72	2	4608	1650
	K _{IB3}	207	33	415	2	782	184
	K _{IB4}	200	32	205	2	1299	311
	K _{IC1}	226	55	163	6	149	18
	K _{IC2}	47	8	217	2	1251	310
	K _{IC3}	195	30	208	2	539	55
	K _{IC4}	131	22	162	2	756	181
SACO-MOCHILA - 2º MÉTODO	K _{A1}	24	2	191	2	39	5
	K _{A2}	174	15	443	2	275	41
	K _{A3}	82	1	974	2	314	45
	K _{A4}	115	2	593	2	268	18
	K _{B1}	46	4	1357	46	111	15
	K _{B2}	69	6	638	2	265	40
	K _{B3}	48	1	256	2	297	24
	K _{B4}	80	1	682	2	563	60
	K _{C1}	29	3	29	2	112	9
	K _{C2}	202	20	1126	22	218	16
	K _{C3}	46	3	1269	2	683	57
	K _{C4}	121	10	733	2	1301	160

Tabela 2.3: (continuação)

	PROBLEMA	ENUMERATIVO		COMPLEMENTAR DE PESQUISA EM ÁRVORE		SEQUENCIAL COMPLEMENTAR	
		NIT	NOT	NIT	NOT	NIT	NOT
SACO-MOCHILA - 3º MÉTODO	K _{A1}	31	5	328	27	31	1
	K _{A2}	600	59	3457	113	72	1
	K _{A3}	385	15	1516	3	1857	242
	K _{A4}	812	38	1102	2	3386	370
	K _{B1}	13	2	34	2	63	12
	K _{B2}	53	4	154	2	86	6
	K _{B3}	1003	81	726	2	849	75
	K _{B4}	413	20	3864	8	574	34
	K _{C1}	12	1	31	1	293	101
	K _{C2}	24	1	72	1	3502	790
	K _{C3}	103	6	209	2	5581	560
	K _{C4}	480	30	343	2	5447	281

Comparando os resultados obtidos pelos algoritmo sequencial complementar e método enumerativo paramétrico conclui-se que este segundo processo apresenta um esforço superior na obtenção da solução óptima relativamente ao primeiro. Note-se que, apesar de ambos asentarem na mesma filosofia de resolução, o algoritmo enumerativo paramétrico necessita de resolver em cada nó o problema linear da relaxação complementar, pelo que a existência de um intervalo de variação muito grande entre o limite inferior e o limite superior em cada nó conduz a um grande aumento na dimensão da árvore de pesquisa. Da experiência efectuada constata-se que a relaxação linear em cada nó não conduz a bons limites inferiores, pois cerca de 51% das iterações usadas pelo algoritmo dizem respeito à resolução dos problemas relaxados. A inclusão de um processo que melhore os limites inferiores deverá beneficiar o desempenho do algoritmo e torná-lo muito mais competitivo com o algoritmo sequencial complementar. Como veremos no próximo capítulo, em outras aplicações do *MPEC*, a existência de melhores limites inferiores para o *MPEC* torna o método enumerativo paramétrico muito mais eficiente na prática.

Uma segunda experiência computacional foi efectuada com os problemas *LCP* NP-difíceis associados a problemas saco-mochila. Tal como foi referido no capítulo 1, a solução do *LCP* requer em geral o cálculo do mínimo global do programa quadrático cuja função objectivo é definida por $z^T w$. O valor óptimo é assim conhecido, já que o *LCP* tem solução se e só se existir um mínimo global com valor óptimo nulo. Na Tabela 2.3 apresenta-se a comparação de resultados obtidos pela resolução do problema utilizando o método enumerativo, o algoritmo complementar de pesquisa em árvore e o algoritmo sequencial complementar. Neste caso o algoritmo enumerativo paramétrico não foi usado, porque os problemas lineares relaxados são ilimitados inferiormente.

Uma primeira análise dos resultados da Tabela 2.3 indica que o método sequencial complementar é de todos o que apresenta pior comportamento na resolução dos problemas em causa. O algoritmo complementar de ramificação em árvore é neste caso mais eficiente, pois termina prematuramente a pesquisa ao encontrar um limite superior igual ao limite inferior nulo fornecido inicialmente. Em média este último algoritmo necessitou de 19 nós para encontrar a solução óptima, enquanto o algoritmo sequencial complementar teve de percorrer 263 nós para atingir o mesmo objectivo.

A análise de desempenho do método enumerativo permite concluir que é um processo eficiente e eficaz na resolução de *LCPs*. Em termos médios, o método enumerativo necessitou de 211 iterações para obter a solução do *LCP*, enquanto que os métodos complementar de pesquisa em árvore e sequencial complementar utilizaram 1021 e 1159 iterações respectivamente.

Embora não se apresentem na tabela os tempos de execução dos problemas salienta-se que são bastante reduzidos, com um tempo máximo obtido de 3.1 segundos.

Como conclusão final dos resultados desta experiência, podemos afirmar que *LCPs* NP-difíceis podem ser resolvidos eficientemente usando o método enumerativo. O método complementar de pesquisa em árvore tem resultados bastante promissores, enquanto que o algoritmo sequencial complementar parece ser menos interessante neste caso. Pensamos que será interessante desenvolver técnicas eficientes para limites inferiores de modo a tornar mais interessante a resolução de *LCPs* através do *MPEC* equivalente. As outras formulações *MPEC* do *LCP* apresentadas no capítulo anterior poderão então ser exploradas devidamente, de forma a poder usar o algoritmo enumerativo paramétrico para a resolução desses problemas.

Capítulo 3

Programas Lineares Inteiros

Neste capítulo aborda-se a resolução de problemas de programação inteira 0-1 através da sua equivalência a *MPECs*.

3.1 Formulação *MPEC* de Programas Inteiros

O problema de Programação Inteira Mista (*MIP*) pode ser escrito na forma

$$\begin{aligned} & \text{Minimizar} && c^T x + d^T y \\ & \text{sujeito a} && Ax + By = b \\ & && x_i \in \{0, 1\}, \quad i = 1, \dots, n \\ & && y \geq 0. \end{aligned} \tag{3.1}$$

Para transformar este problema num *MPEC*, consideremos as variáveis

$$w_i = 1 - x_i, \quad \forall i = 1, \dots, n.$$

Então é fácil de concluir que para todo $i = 1, \dots, n$,

$$x_i \in \{0, 1\} \iff x_i \geq 0, w_i \geq 0, x_i w_i = 0, w_i = 1 - x_i.$$

Considerando esta equivalência, então obtemos o seguinte *MPEC*

$$\begin{aligned} & \text{Minimizar} && c^T x + d^T y \\ & \text{sujeito a} && Ax + By = b \\ & && w = e - x \\ & && x, w, y \geq 0 \\ & && x^T w = 0, \end{aligned} \tag{3.2}$$

com e um vector de \mathbb{R}^n de componentes unitárias. Este problema pode-se escrever na forma mais simples, idêntico ao sugerido em [Vicente et al., 1996]

$$\begin{aligned}
& \text{Minimizar} && c^T x + d^T y \\
& \text{sujeito a} && Ax + By = b \\
& && 0 \leq x \leq e \\
& && y \geq 0 \\
& && x^T(e - x) = 0.
\end{aligned} \tag{3.3}$$

Seguidamente iremos mostrar que a formulação de Dois Níveis de um Programa Inteiro 0-1 apresentada em [Audet et al., 1997] se reduz à formulação *MPEC* (3.2). Segundo os autores, o programa inteiro (3.1) é equivalente ao seguinte Programa de Dois Níveis

$$\begin{aligned}
& \min_{x,y,u} && c^T x + d^T y \\
& \text{sujeito a} && Ax + By = b \\
& && 0 \leq x \leq e \\
& && y \geq 0 \\
& && u = 0 \\
& && \arg \min_u -e^T u \\
& \text{sujeito a} && u \leq x \\
& && u \leq e - x.
\end{aligned}$$

De modo a reduzir o número de restrições de igualdade podemos substituir $u = 0$ pelas restrições equivalentes

$$e^T u = 0 \quad \text{e} \quad u \geq 0,$$

onde e é um vector de \mathbb{R}^n de componentes unitárias. Tal como fizemos no capítulo 1, podemos reduzir esse programa de dois níveis a um *MPEC*, substituindo o problema do segundo nível pelas respectivas condições de *Karush-Kuhn-Tucker*. Desse modo obtemos o seguinte *MPEC*

$$\begin{aligned}
& \text{Minimizar} && c^T x + d^T y \\
& \text{sujeito a} && Ax + By = b \\
& && 0 \leq x \leq e \\
& && y \geq 0, u \geq 0 \\
& && v + u - x = 0 \\
& && w + u + x = e \\
& && \alpha + \beta = e \\
& && e^T u = 0 \\
& && v, w, \alpha, \beta \geq 0 \\
& && v^T \alpha = \beta^T w = 0
\end{aligned}$$

que foi usado em [Júdice et al., 2006a] para a resolução de um problema de estruturas com variáveis discretas. Para simplificar este problema, começemos por estabelecer o seguinte resultado.

Teorema 3.1 *Se $v_i \geq 0$ e $w_i \geq 0$, então*

$$\alpha_i + \beta_i = 1, v_i \alpha_i = \beta_i w_i = 0, \alpha_i \geq 0, \beta_i \geq 0 \iff v_i w_i = 0.$$

Demonstração: Para demonstrar a implicação \Rightarrow , tem-se

$$v_i > 0 \Rightarrow \alpha_i = 0 \Rightarrow \beta_i = 1 \Rightarrow w_i = 0$$

$$w_i > 0 \Rightarrow \beta_i = 0 \Rightarrow \alpha_i = 1 \Rightarrow v_i = 0.$$

Para estabelecer a implicação inversa, notemos que $v_i w_i = 0$ implica $v_i = 0$ ou $w_i = 0$. Se $v_i = 0$, então $\alpha_i = 1, \beta_i = 0$ satisfazem as condições $v_i \alpha_i = \beta_i w_i = 0, \alpha_i + \beta_i = 1, \alpha_i \geq 0, \beta_i \geq 0$. De outro modo $w_i = 0$ e as condições verificam-se com $\alpha_i = 0, \beta_i = 1$. \square

Além disso podemos substituir o vector u por $(x - v)$ e obter o seguinte *MPEC*

$$\begin{aligned} & \text{Minimizar} && c^T x + d^T y \\ & \text{sujeito a} && Ax + By = b \\ & && 0 \leq x \leq e \\ & && y \geq 0 \\ & && x \geq v \\ & && w = e + v - 2x \\ & && e^T x - e^T v = 0 \\ & && w \geq 0, v \geq 0 \\ & && v^T w = 0 \end{aligned} \tag{3.4}$$

e verifica-se o seguinte resultado.

Teorema 3.2 *Os *MPECs* (3.2) e (3.4) são equivalentes.*

Demonstração: Basta notar as definições dos dois *MPECs* e a seguinte equivalência

$$e^T x - e^T v = 0 \wedge x \geq v \iff x = v.$$

\square

3.2 Algoritmos *MPEC* para Programação Inteira

Na última secção mostrámos que um programa inteiro misto com variáveis 0-1 é equivalente ao *MPEC* (3.2). A matriz associada às variáveis x_i é semi-definida negativa e portanto

o *GLCP* que constitui o seu conjunto de restrições é um problema NP-difícil. Por isso o método complementar de pesquisa em árvore discutido no capítulo anterior não pode ser usado para a determinação do mínimo global desse *MPEC*. Contudo os dois restantes algoritmos, sequencial complementar e enumerativo paramétrico, podem ser úteis neste tipo de aplicação do *MPEC*. Dada a forma simplificada deste *MPEC*, apresentada em (3.3), é possível desenvolver esses processos de um modo mais simples sem usar explicitamente as variáveis complementares w_i . Esses processos serão descritos nesta secção.

(I) Algoritmo Sequencial Complementar

Conforme foi discutido no capítulo 2, este algoritmo resolve um *MPEC* a partir da determinação de uma sucessão de pontos estacionários com valores estritamente decrescentes da função objectivo. Se o *MPEC* é difícil de resolver, então um método enumerativo é usado para determinar uma solução do *GLCP* que constitui o seu conjunto de restrições. A implementação do método enumerativo descrito no capítulo 2 ao *MPEC* (3.2) é muito simples de fazer, bastando para o efeito efectuar algumas alterações que são discutidas a seguir.

- (i) O programa quadrático associado a cada nó visitado pelo algoritmo tem a forma

$$\begin{aligned} \text{Minimizar} \quad & x^T(e - x) \\ \text{sujeito a} \quad & Ax + By = b \\ & 0 \leq x \leq e \\ & y \geq 0 \\ & x_i = 0, i \in I \\ & x_j = 1, j \in J \\ & c^T x + d^T y \leq \lambda_k, \end{aligned}$$

com I e J subconjuntos disjuntos de $\{1, \dots, n\}$.

- (ii) A ramificação é feita de acordo com a dicotomia $x_i = 0 \vee x_i = 1$.

- (iii) A escolha do nó é feita a partir de

$$\bar{x}(l)^T(e - \bar{x}(l)) = \min \{ \bar{x}(r)^T(e - \bar{x}(r)) : r \text{ é nó aberto} \}$$

onde $\bar{x}(r)$ é o ponto estacionário associado ao nó r .

- (iv) Para escolha da variável x_i a ramificar, utiliza-se o seguinte critério:

$$\bar{x}_r(1 - \bar{x}_r) = \max \{ \bar{x}_i(1 - \bar{x}_i) : i \notin I \cup J \},$$

com I, J os conjuntos de índices das variáveis marcadas nesse nó e \bar{x} o ponto estacionário do programa quadrático a si associado.

- (v) O algoritmo termina com uma solução do *GLCP* quando se obtiver num nó um ponto estacionário com valor nulo da função $x^T(e - x)$. Uma outra forma de terminar o método consiste em esgotar a lista de nós abertos sem obter um ponto estacionário com valor nulo para $x^T(e - x)$. Nesse caso o *GLCP* não tem solução.

Um outro processo incluído no algoritmo sequencial complementar é o método complementar de restrições activas. Como referimos no capítulo 2, este algoritmo necessita de uma solução do *GLCP* para se poder utilizar. Qualquer uma dessas soluções (\bar{x}, \bar{w}) com $\bar{w} = e - \bar{x}$ satisfaz

$$\bar{x}_i + \bar{w}_i = 1 \quad \forall i = 1, \dots, n$$

e portanto as variáveis complementares não básicas nunca podem ser candidatas a aumentar o seu valor. Para programas inteiros puros que não contêm variáveis y_i , toda a solução do *GLCP* é assim um ponto estacionário do *MPEC*. Para programas mistos o algoritmo complementar de restrições activas reduz-se a um método simplex modificado. Em cada iteração desse processo uma variável não básica $y_i = 0$ passará a básica se o respectivo custo reduzido for negativo e trocará com uma outra variável básica. Se todos os coeficientes de custos reduzidos associados às variáveis não básicas y_i são não negativos, então o algoritmo termina com um ponto estacionário do *MPEC*.

Após as descrições sumárias dos dois algoritmos, enumerativo e complementar de restrições activas, é possível apresentar os passos do algoritmo sequencial complementar na seguinte forma

ALGORITMO SEQUENCIAL COMPLEMENTAR

Passo 0: *Inicialização*

Seja $k = 0$ e $\lambda_0 = c^T \bar{x} + d^T \bar{y}$, com (\bar{x}, \bar{y}) uma solução inteira obtida por um processo heurístico ($\lambda_0 = +\infty$ se tal solução não for conhecida).

Passo 1: *Determinação de uma solução complementar*

Determine uma solução do *QP*

$$\begin{aligned} QP(\lambda_k) \quad & \text{Minimizar} \quad x^T(e - x) \\ & \text{sujeito a} \quad Ax + By = b \\ & \quad \quad \quad 0 \leq x \leq e \\ & \quad \quad \quad y \geq 0 \\ & \quad \quad \quad c^T x + d^T y \leq \lambda_k \end{aligned}$$

usando o método enumerativo.

Passo 2: *Determinação de um ponto estacionário*

Se o valor óptimo do $QP(\lambda_k)$ é nulo, então determine um ponto estacionário (\bar{x}, \bar{y}) do *MPEC*

usando o algoritmo de restrições activas complementar (se o programa inteiro é puro, a solução óptima do $QP(\lambda_k)$ é um ponto estacionário). Faça $\lambda_k = c^T \bar{x} + d^T \bar{y}$ e actualize

$$\lambda_{k+1} = \lambda_k - \gamma |\lambda_k|,$$

com $\gamma > 0$ uma tolerância para zero. Vá para Passo 1 com $k \leftarrow k + 1$.

Passo 3: *Optimalidade*

Se o valor óptimo do $QP(\lambda_k)$ é positivo, então o ponto estacionário do $MPEC$ obtido na iteração $(k - 1)$ é uma solução óptima do programa inteiro e termine.

Tal como anteriormente, o algoritmo sequencial complementar é essencialmente baseado na determinação de pontos estacionários de programas quadráticos e pode, por isso, ser implementado usando o código MINOS de restrições activas para programação quadrática.

(II) Algoritmo Enumerativo Paramétrico

Tal como no caso do algoritmo sequencial complementar, é fácil descrever os passos do algoritmo enumerativo paramétrico. O processo utiliza os programas linear relaxado associado ao programa inteiro e o programa quadrático $QP(\lambda_k)$ introduzido na descrição do método sequencial. Os passos do processo são apresentados a seguir, onde, como anteriormente, se assume que o programa linear relaxado tem solução óptima.

ALGORITMO ENUMERATIVO PARAMÉTRICO

Passo 0: *Inicialização*

Seja λ_1 um número positivo grande e a lista de nós em aberto $\mathcal{L} = \{1\}$.

Determine a solução óptima do Programa Linear relaxado (x^*, y^*) e faça

$$\text{VAL}(1) = c^T x^* + d^T y^*.$$

Passo 1: *Avaliação da optimalidade*

Se $\mathcal{L} = \emptyset$, então o último ponto estacionário do $MPEC$ obtido é a solução óptima do programa inteiro, correspondendo a UB .

Passo 2: *Escolha do nó a ramificar*

Determine o nó $t \in \mathcal{L}$, tal que

$$\text{VAL}(t) = \min\{\text{VAL}(i) : i \in \mathcal{L}\}.$$

Passo 3: *Determinação de Ponto estacionário do GLCP*

Determine o ponto estacionário (\bar{x}, \bar{y}) de

$$\begin{aligned} & \text{Minimizar} && x^T(e - x) \\ & \text{sujeito a} && Ax + Gy = b \\ & && 0 \leq x \leq e \\ & && y \geq 0 \\ & && x_i = 0, i \in I \\ & && x_j = 1, j \in J \\ & && c^T x + d^T y \leq \lambda_k, \end{aligned}$$

com I e J os conjuntos dos índices das variáveis marcadas nesse nó. Para esse efeito utilize o método de restrições activas com ponto inicial a solução associada a $\text{VAL}(t)$.

Passo 4: *Determinação de Ponto estacionário do MPEC*

Se o ponto estacionário (\bar{x}, \bar{y}) é uma solução complementar, ou seja, se $\bar{x}^T(e - \bar{x}) = 0$, aplique o método de restrições activas complementar com ponto inicial (\bar{x}, \bar{y}) e determine um ponto estacionário (\tilde{x}, \tilde{y}) do Programa Inteiro. Actualize

$$\begin{aligned} UB &= c^T \tilde{x} + d^T \tilde{y} \\ \lambda_k &= UB - \gamma |UB| \\ \mathcal{L} &= \mathcal{L} \setminus \{t \in \mathcal{L} : \text{VAL}(t) \geq UB\} \end{aligned}$$

com $\gamma > 0$ e pequeno e volte ao Passo 1.

Passo 5: *Escolha da variável de ramificação*

Se o ponto estacionário (\bar{x}, \bar{y}) não é uma solução complementar, escolha s a partir de

$$\bar{x}_s(1 - \bar{x}_s) = \max\{\bar{x}_i(1 - \bar{x}_i) : 0 < \bar{x}_i < 1\}.$$

Passo 6: *Determinação do número de ramos*

Considere os nós associados a $x_s = 0$ e $x_s = 1$ e resolva os programas lineares associados

$$\begin{aligned} & \text{Minimizar} && c^T x + d^T y \\ & \text{sujeito a} && Ax + Gy = b \\ & && 0 \leq x \leq e, y \geq 0 \\ & && x_i = 0, i \in I \\ & && x_j = 1, j \in J, \end{aligned}$$

com $I = I \cup \{s\}$ e $J = J \cup \{s\}$ respectivamente. Para qualquer um desses programas lineares admissíveis faça

$$\text{VAL}(|\mathcal{L}| + r) = c^T x^* + d^T y^*,$$

com $r = 1, 2$ e (x^*, y^*) a solução óptima do LP respectivo. Volte ao Passo 1, com

$$\mathcal{L} = \mathcal{L} \setminus \{s\} \cup \{|\mathcal{L}| + r : r \in \{0, 1, 2\}\}$$

dependendo do número de programas lineares admissíveis.

3.3 Experiência Computacional

(I) Descrição dos problemas

A metodologia descrita na secção anterior foi aplicada a quatro tipos de programas inteiros binários, convertidos em problemas *MPEC* através da formulação (3.3) e que se descrevem seguidamente.

Problemas Tipo I. *Problemas de Árvore de Suporte de Custo Mínimo com Restrição de Capacidade*

O problema de Árvore de Suporte de Custo Mínimo com Restrição de Capacidade (*Cost Minimum Support Tree*), que se designará por CMST, pode ser definido sobre um grafo cujos nodos representam conjuntamente os terminais e a unidade central (raiz) e cujas arestas caracterizam as linhas de transmissão.

Se considerarmos o grafo completo não orientado $G=(V,E)$, onde $V = \{0, 1, \dots, n\}$ é o conjunto dos nós terminais incluindo o nó 0 que constitui a raiz e $E = \{(i, j) : i, j \in V, i < j\}$ define o conjunto das arestas entre os nodos de V , podemos dizer que cada aresta (i, j) em E tem associado um peso ou custo não negativo designado por c_{ij} .

Um subconjunto minimal de arestas de E que garantem a conexidade entre todos os nodos do grafo define uma *árvore* em G , subgrafo sem ciclos, designado por *árvore de suporte* T sobre G . O custo da árvore T , designado por $C(T)$, obtém-se pela seguinte expressão

$$C(T) = \sum_{(i,j) \in T} c_{ij}.$$

Se em T se eliminarem todas as arestas incidentes na raiz, o novo grafo constitui uma floresta, onde cada componente conexa é um subconjunto de nodos e arestas que definem uma subárvore de T . Desta definição resulta que cada subárvore tem uma única aresta incidente na raiz. O problema CMST pode assim ser definido pela determinação de uma árvore de suporte T de custo mínimo, tal que o número de nodos (excepto a raiz) em qualquer subárvore de T não seja superior ao valor da capacidade Q .

Para a formulação do problema considerem-se os dois conjuntos de:

- variáveis binárias, definidas sobre o grafo orientado

$$x_{ij} = \begin{cases} 1 & \text{se o arco } (i, j) \text{ está na solução} \\ 0 & \text{caso contrário} \end{cases} \quad i = 0, \dots, n \quad \text{e} \quad j = 1, \dots, n$$

- variáveis adicionais de fluxo, não negativas

$$y_{ij} = \text{quantidade de fluxo que sai da raiz e passa pelo arco } (i, j), \\ \text{com } i = 0, \dots, n \text{ e } j = 1, \dots, n.$$

A formulação do problema CMST, que se apresenta a seguir, foi proposta em [Gavish, 1983]

$$\begin{array}{ll}
 \text{Minimizar} & \sum_{i=0}^n \sum_{j=1}^n c_{ij} x_{ij} \\
 \text{sujeito a} & \sum_{i=0}^n x_{ij} = 1 \quad j = 1, \dots, n, \quad i \neq j \\
 & \sum_{i=0}^n y_{ij} - \sum_{i=1}^n y_{ji} = 1 \quad j = 1, \dots, n, \quad i \neq j \\
 & \left. \begin{array}{l} x_{ij} \leq y_{ij} \leq (Q - d_i) x_{ij} \\ x_{ij} \in \{0, 1\} \\ y_{ij} \geq 0 \end{array} \right\} \quad i = 0, \dots, n; \quad j = 1, \dots, n, \quad i \neq j,
 \end{array}$$

com $d_0 = 0$ e $d_i = 1$, $\forall i = 1, \dots, n$. Para simplificação da formulação não se incluem as variáveis x_{ii} e y_{ii} com $i = 1, \dots, n$. Foram testadas algumas instâncias de pequena dimensão para o problema CMST, em que $C_{(n+1) \times (n+1)}$ é uma matriz simétrica que define a distância Euclideana entre as coordenadas cartesianas dos nodos i e j . A localização da raiz no centro ou num dos vértices de uma rede quadrada caracteriza duas classes de instâncias distintas, tendo sido testados os problemas cujas características se apresentam na Tabela 3.1.

Tabela 3.1: Características dos problemas CMST

PROBLEMA	LOCALIZAÇÃO		
	DA RAIZ	n	Q
CMST ₁	Centro	5	3
CMST ₂	Centro	5	4
CMST ₃	Vértice	5	4
CMST ₄	Centro	7	5

Problemas Tipo II. Problemas de Localização com Capacidades

Para a sua descrição considerem-se as seguintes variáveis associadas

$$y_j = \begin{cases} 1 & \text{se o serviço é aberto no local } j \\ 0 & \text{caso contrário.} \end{cases}$$

x_{ij} = fracção da procura que o cliente i recebe do serviço j .

c_{ij} = custo de servir na totalidade o cliente i a partir do serviço j .

u_i = procura do cliente i .

g_j = capacidade do serviço j .

f_j = custo de abrir o serviço j .

com $i = 1, \dots, m$ e $j = 1, \dots, n$.

A formulação do problema apresenta a seguinte forma [Cornuéjols et al., 1991]

$$\begin{aligned}
 & \text{Minimizar} && \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n f_j y_j \\
 & \text{sujeito a} && \sum_{j=1}^n x_{ij} = 1 && i = 1, \dots, m \\
 & && \sum_{i=1}^m u_i x_{ij} \leq g_j y_j && j = 1, \dots, n \\
 & && y_j \in \{0, 1\} && j = 1, \dots, n \\
 & && x_{ij} \geq 0 && i = 1, \dots, m; \quad j = 1, \dots, n.
 \end{aligned}$$

Os problemas gerados, designados por LOC., foram obtidos atribuindo os seguintes parâmetros na formulação anterior

$$c_{ij} = \frac{i}{1000} \quad i = 1, \dots, m; \quad j = 1, \dots, n.$$

$$f_j = \frac{j}{10} \quad j = 1, \dots, n.$$

$$u_i = \frac{500}{i} \quad i = 1, \dots, m.$$

$$g_j = \frac{4\alpha}{n}.$$

com

$$\alpha = \sum_{i=1}^m u_i$$

e m e n a tomarem os valores apresentados na Tabela 3.2.

Tabela 3.2: *Dimensão dos problemas LOC*

PROBLEMA	m	n
LOC ₁	50	50
LOC ₂	50	75
LOC ₃	50	100
LOC ₄	100	100
LOC ₅	100	120
LOC ₆	100	150

Problemas Tipo III. Problemas Saco-Mochila

Os problemas testados encontram-se descritos em [Cornuéjols et al., 2003] e apresentam a

seguinte formulação

$$\begin{aligned} & \text{Maximizar} && \sum_{j=1}^n p_j x_j \\ & \text{sujeito a} && \sum_{j=1}^n w_j x_j \leq c \\ & && x_j \in \{0, 1\} \quad j = 1, \dots, n. \end{aligned}$$

Os parâmetros p_j e w_j foram gerados aleatoriamente com distribuição uniforme no intervalo $[1, 1000]$ e a constante c satisfaz a equação

$$c = 0.5 \sum_{j=1}^n w_j.$$

As instâncias testadas apresentam-se na Tabela 3.3.

Tabela 3.3: *Dimensão dos problemas KNAP*

PROBLEMA	n
KNAP ₁	10
KNAP ₂	20
KNAP ₃	30
KNAP ₄	40
KNAP ₅	50
KNAP ₆	100
KNAP ₇	200
KNAP ₈	300
KNAP ₉	400
KNAP ₁₀	500

Problemas Tipo IV. *Problemas da colecção Miplib*

São problemas utilizados frequentemente por especialistas da comunidade científica para testar algoritmos de resolução de programas inteiros [Bixby et al., 1992], que se designam por MIP. As características do conjunto de problemas utilizado da colecção *Miplib* encontram-se descritas na Tabela 3.4, sendo M o número de linhas do problema e nx e ny o número de variáveis inteiras e contínuas respectivamente.

(II) Resultados Computacionais

Tal como anteriormente, em todas as tabelas desta secção, NIS e NOS representam o número de iterações e nodos utilizados pelo método até à obtenção da solução óptima, sendo as restantes colunas do quadro relativos aos parâmetros para a resolução total do problema. De notar que a utilização dos métodos sequencial complementar e enumerativo paramétrico

Tabela 3.4: *Características dos problemas MIP*

PROBLEMA	nx	ny	M
MIP ₁	9	0	13
MIP ₂	27	0	118
MIP ₃	15	0	36
MIP ₄	21	46	45
MIP ₅	100	80	24
MIP ₆	48	0	25
MIP ₇	201	0	133
MIP ₈	40	0	23
MIP ₉	33	0	16
MIP ₁₀	48	48	62
MIP ₁₁	74	62	300
MIP ₁₂	159	1	96
MIP ₁₃	58	1	39
MIP ₁₄	82	1	54
MIP ₁₅	100	0	21
MIP ₁₆	27	0	20

para a resolução dos problemas saco-mochila obriga à escolha de um parâmetro de corte γ bastante reduzido, devido aos valores elevados dos coeficientes da função objectivo. Esta escolha tem efeito sobre a progressão do algoritmo para a solução óptima. No sentido de contornar este problema, foi também testada uma outra versão dos problemas em que os valores da função objectivo foram escalonados, continuando a ter distribuição uniforme mas agora no intervalo $[0, 1]$. Pelo facto de não se terem verificado melhorias no desempenho dos algoritmos, estes resultados não serão apresentados.

A primeira experiência computacional foi realizada com o algoritmo sequencial complementar e o seu comportamento é ilustrado na Tabela 3.5. Os resultados mostram que o algoritmo foi eficiente na determinação da solução óptima da maioria dos problemas, tendo utilizado um número pequeno de iterações e de nodos pesquisados.

O método enumerativo paramétrico utilizado para a obtenção dos resultados da Tabela 3.6 revelou ter um desempenho consideravelmente inferior ao processo sequencial complementar em oito dos exemplos testados. Em termos médios essa redução de desempenho na obtenção da solução óptima do problema face ao esforço total do processo é da ordem dos 35%. Analisando o caso contrário, verifica-se que em 18 dos exemplos testados houve uma melhoria no esforço para a obtenção da solução óptima com a aplicação do método enumerativo paramétrico da ordem média dos 52%, tendo-se verificado em dois dos casos melhoria acima dos 90%. De notar ainda que a diminuição média do número de iterações gastas na obtenção da solução óptima neste último conjunto de exemplos é da ordem das 42500 iterações, ao passo que nos casos em que o método sequencial complementar tem um comportamento

Tabela 3.5: *Resolução dos problemas utilizando o algoritmo Sequencial Complementar*

	PROBLEMA	NIS	NOS	NIT	NOT	T
CMST	CMST ₁	37	1	78	5	3.11E-02
	CMST ₂	463	30	637	49	0.14
	CMST ₃	214	28	249	37	7.8E-02
	CMST ₄	71	2	17225	1685	2.44
LOCALIZAÇÃO	LOC ₁	38249	6171	39710	6444	17.81
	LOC ₂	195	1	683	179	0.66
	LOC ₃	249	1	249	1	0.34
	LOC ₄	436	1	436	1	0.84
	LOC ₅	528	1	528	1	1.06
	LOC ₆	147639	18141	147639	18141	712.08
SACO-MOCHILA	KNAP ₁	24	3	61	24	6.3E-02
	KNAP ₂	28	1	87	32	4.6E-02
	KNAP ₃	809	397	920	454	0.15
	KNAP ₄	544	311	634	386	0.16
	KNAP ₅	1256	533	1338	572	0.28
	KNAP ₆	3731	1788	5256	2495	0.84
	KNAP ₇	6403	2382	10073	3825	1.64
	KNAP ₈	12895	4989	14988	5820	3.02
	KNAP ₉	403	1	947	253	0.17
	KNAP ₁₀	67705	24361	74295	26592	17.99
MIPLIB	MIP ₁	7	1	96	42	7.79E-02
	MIP ₂	283	8	66477	10147	10.11
	MIP ₃	24	1	914	253	0.13
	MIP ₄	2434	364	4566	687	0.64
	MIP ₅	199	1	52903	3655	4.21
	MIP ₆	72907	9933	85253	11674	7.35
	MIP ₇	73116	839	273450	3210	39.60
	MIP ₈	3428	113	4932	184	0.31
	MIP ₉	25502	5593	44193	10702	4.84
	MIP ₁₀	4161	258	13403	985	1.23
	MIP ₁₁	6732	139	28894	876	7.55
	MIP ₁₂	3528	48	58938	923	7.67
	MIP ₁₃	346	6	1689	77	0.15
	MIP ₁₄	1205	41	13136	586	1.19
	MIP ₁₅	582495	39536	582495	39536	38.95
	MIP ₁₆	1075	72	3973	451	0.36

melhor que o método enumerativo paramétrico a diminuição média de iterações é de apenas 21000.

Tabela 3.6: *Resolução dos problemas utilizando o algoritmo Enumerativo Paramétrico*

	PROBLEMA	NIS	NOS	NIT	NOT	T
CMST	CMST ₁	37	1	81	5	3.98E-02
	CMST ₂	125	5	194	15	3.09E-02
	CMST ₃	79	5	80	7	0.0
	CMST ₄	172	12	23312	1891	4.13
LOCALIZAÇÃO	LOC ₁	2281	155	2491	217	2.01
	LOC ₂	195	1	1756	179	1.37
	LOC ₃	249	1	249	1	0.21
	LOC ₄	436	1	436	1	0.68
	LOC ₅	528	1	528	1	1.13
	LOC ₆	136570	12284	136570	12286	695.20
SACO-MOCHILA	KNAP ₁	25	6	61	16	3.4E-02
	KNAP ₂	57	13	117	29	1.65E-02
	KNAP ₃	149	46	157	48	3.11E-02
	KNAP ₄	161	59	178	63	4.6E-02
	KNAP ₅	138	28	158	37	1.63E-02
	KNAP ₆	1423	681	1462	698	0.42
	KNAP ₇	3171	1413	3173	1421	0.87
	KNAP ₈	1158	408	2358	970	0.83
	KNAP ₉	403	1	893	253	0.25
	KNAP ₁₀	6419	3049	7645	3495	3.49
MIPLIB	MIP ₁	51	5	168	49	0.12
	MIP ₂	655	19	87918	9463	17.84
	MIP ₃	24	1	1550	243	0.23
	MIP ₄	1820	127	3066	353	0.40
	MIP ₅	262	1	104761	3831	7.51
	MIP ₆	12338	900	12522	1439	1.92
	MIP ₇	199718	1461	199766	1541	34.18
	MIP ₈	195	5	1790	83	0.12
	MIP ₉	12594	1907	13733	2142	1.69
	MIP ₁₀	3258	179	4191	482	0.90
	MIP ₁₁	6848	94	16781	447	8.41
	MIP ₁₂	35218	328	64072	883	8.5
	MIP ₁₃	1867	54	1867	65	0.18
	MIP ₁₄	7690	210	18480	732	1.84
	MIP ₁₅	22761	1027	22761	1027	1.79
	MIP ₁₆	879	31	4144	391	0.50

A comparação entre o método enumerativo paramétrico e o método sequencial complementar mostra ainda que o último processo tem em geral um melhor desempenho para pequenas dimensões. No entanto, à medida que a dimensão aumenta, o método enumerativo paramétrico torna-se mais eficiente na determinação da solução óptima. Para confirmar este resultado, apresenta-se na Tabela 3.7 uma comparação dos dois processos, em relação ao número de iterações gastas para a obtenção da solução óptima face à dimensão do problema.

Tabela 3.7: *Desempenho dos algoritmos para os problemas inteiros*

	ENUMERATIVO	
	SLCP	PARAMÉTRICO
MELHOR	0.05	0.05
MÉDIA	232.59	47.39
PIOR	5824.95	597.96

Apresentam-se ainda na Tabela 3.8 os resultados da resolução destes problemas de teste com o programa *Cplex* [ILOG, 2002], por forma a comparar o desempenho dos algoritmos complementares com o programa comercial. Da observação dos resultados é evidente o melhor desempenho do programa *Cplex*. É de notar que este algoritmo utiliza técnicas de corte eficientes para a determinação de limites inferiores que não foram incorporados nos métodos complementares estudados neste capítulo.

Tabela 3.8: *Resolução dos problemas utilizando o programa Cplex*

PROBLEMA		NIT	NOT	T	FOBJ
CMST	CMST ₁	36	0	0,05	92
	CMST ₂	104	15	0,09	281
	CMST ₃	27	0	0,05	120
	CMST ₄	21057	2426	1,66	140
LOCALIZAÇÃO	LOC ₁	576	55	0.56	10.375
	LOC ₂	566	28	0.47	20.275
	LOC ₃	520	0	0.22	33.775
	LOC ₄	760	0	0.66	37.55
	LOC ₅	796	0	2.14	51.55
	LOC ₆	18610	3999	88.28	79.15
SACO-MOCHILA	KNAP ₁	2	0	0.02	-3595.12
	KNAP ₂	2	0	0.0	-8286.7
	KNAP ₃	5	0	0.0	-12011.83
	KNAP ₄	4	0	0.02	-15951.73
	KNAP ₅	3	0	0.0	-19133.90
	KNAP ₆	45	37	0.02	-39632.15
	KNAP ₇	119	93	0.03	-79312.98
	KNAP ₈	134	131	0.05	-120594.98
	KNAP ₉	1	0	0.02	-163489.65
	KNAP ₁₀ ⁽¹⁾	56	50	0.06	-202401.02

⁽¹⁾ A solução óptima obtida não é a solução óptima global, utilizando os parâmetros por defeito do *CPLEX*.

O valor óptimo da função objectivo é -202403.07.

Tabela 3.8: (continuação)

	PROBLEMA	NIT	NOT	T	FOBJ
MIPLIB	MIP ₁	35	8	0.00	5.0
	MIP ₂	12382	3833	0.51	18.0
	MIP ₃	314	91	0.02	9.0
	MIP ₄	499	90	0.03	375.0
	MIP ₅	8694	2332	0.36	82.15
	MIP ₆	193	18	0.01	788.263
	MIP ₇	3655	209	0.47	7615.0
	MIP ₈	15	0	0.02	62027.0
	MIP ₉	69	4	0.02	3089.0
	MIP ₁₀	616	72	0.05	280.95
	MIP ₁₁	2044	129	0.22	2984.5
	MIP ₁₂	14335	1556	0.86	3360.0
	MIP ₁₃	231	31	0.03	1690.0
	MIP ₁₄	4721	629	0.23	563.5
	MIP ₁₅	22993	4789	0.66	0.0
	MIP ₁₆	610	114	0.05	34.0

Como conclusão final deste estudo experimental, podemos afirmar que ambos os métodos sequencial complementar e enumerativo paramétrico resolvem todos os programas lineares inteiros de uma forma razoavelmente eficiente. Além disso, o método enumerativo paramétrico parece ser mais interessante do que o método sequencial complementar, particularmente à medida que a dimensão aumenta. A incorporação de técnicas de corte para a determinação de melhores limites inferiores poderá ainda tornar este processo mais eficaz para resolver programas inteiros e assim ser capaz de competir com uma técnica de pesquisa em árvore como o *Cplex*, que não tira partido da complementaridade.

Capítulo 4

Número de Independência de um Grafo

Neste capítulo estudamos a aplicação da Programação Quadrática Não Convexa e Programação Matemática com Restrições de Equilíbrio à determinação do número de independência de um grafo. São discutidas formulações *MPEC* para o problema e a aplicabilidade dos algoritmos referidos anteriormente para a sua resolução. Além disso é introduzido um algoritmo de ramificação e limitação para a resolução de um programa quadrático, baseado na decomposição de uma função quadrática numa diferença de duas funções convexas (DC).

4.1 Definição do Problema

O problema da determinação do conjunto independente máximo de um grafo, do inglês *Maximum Independent Set (MIS)*, é um problema NP-completo [Garey e Johnson, 1983] com um vasto leque de aplicações [Du e Pardalos, 1999, Berge, 1982]. Nesta secção iremos apresentar a definição do problema, assim como a sua relação com um problema de programação linear inteira.

Definição 4.1 *Seja $G = (V, E)$ um grafo, com V e E os conjuntos dos vértices e das arestas respectivamente.*

(i) *Um conjunto independente de G é um conjunto de vértices sem qualquer ligação entre eles. O número de independência $\alpha(G)$ de G é o cardinal do maior conjunto independente de G .*

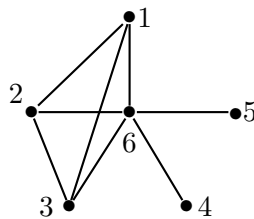
(ii) *Uma clique de G é um conjunto de vértices todos eles ligados entre si, ou equivalen-*

temente, um subgrafo completo de G . O número de clique $\omega(G)$ de G é o cardinal da maior clique de G .

(iii) O número de cobertura por cliques $\theta(G)$ de G é o número mínimo de cliques que cobrem todos os vértices de um grafo.

(iv) O número cromático $\gamma(G)$ de G é o número mínimo de cores que são necessárias para colorir um grafo de modo a que dois vértices adjacentes não apresentem a mesma cor, ou seja, é o número mínimo de conjuntos independentes que cobrem os vértices de um grafo.

Para ilustração destas definições, consideremos o grafo G



Então $\{1, 4, 5\}$ e $\{2, 4, 5\}$ são conjuntos independentes máximos e $\alpha(G)=3$. O conjunto dos vértices $\{1, 2, 3, 6\}$ é a clique de cardinalidade máxima e portanto $\omega(G) = 4$. O número de cobertura por cliques é $\theta(G) = 3$ e corresponde à partição definida pelas cliques $\{1, 2, 3, 6\}, \{4\}$ e $\{5\}$. Finalmente o número cromático de G é $\gamma(G) = 4$ e é obtido através da partição do grafo nos conjuntos independentes $\{1, 4, 5\}$, $\{2\}$, $\{3\}$ e $\{6\}$.

Dado um grafo G , chama-se Grafo Complementar de G , e representa-se por \bar{G} , o grafo constituído pelos mesmos vértices e cujas arestas são as que não pertencem a G . O próximo resultado é consequência das definições apresentadas anteriormente.

Teorema 4.1 *Se \bar{G} é o grafo complementar de G , então*

$$(i) \quad \omega(G) = \alpha(\bar{G}).$$

$$(ii) \quad \alpha(G) \leq \theta(G).$$

$$(iii) \quad \omega(G) \leq \gamma(G).$$

Definição 4.2 *O Poliedro dos conjuntos independentes de $G = (V, E)$ é*

$$P(G) = \text{Invólucro Convexo}\{x \in \{0, 1\}^n : x_i + x_j \leq 1, \forall (i, j) \in E\},$$

com n o número de vértices do grafo.

Devido à definição 4.2, cada vértice do poliedro $P(G)$ corresponde a um conjunto independente do grafo tal que as coordenadas desse elemento são

$$x_i = \begin{cases} 1 & \text{se o vértice está no conjunto independente,} \\ 0 & \text{caso contrário.} \end{cases}$$

Assim sendo, a determinação do conjunto independente máximo corresponde à solução ótima do programa linear inteiro misto

$$\begin{aligned} & \text{Maximizar} && \sum_{i=1}^n x_i \\ & \text{sujeito a} && x_i + x_j \leq 1, \forall (i, j) \in E \\ & && x_i \in \{0, 1\}, \forall i \in V. \end{aligned}$$

4.2 Formulações Quadráticas

De acordo com [Horst e Pardalos, 1995], o valor $\alpha(G)$ do grafo G pode ser determinado através da resolução de um problema quadrático de programação inteira da forma

$$\begin{aligned} & \text{Maximizar} && 2e^T x - x^T (A + I)x \\ & \text{sujeito a} && x \in \{0, 1\}^n, \end{aligned} \tag{4.1}$$

sendo I a matriz identidade de ordem n e A a matriz de adjacência do grafo G . Portanto os elementos a_{ij} de A são definidos por

$$a_{ij} = \begin{cases} 1 & \text{se a aresta } (i, j) \in E \\ 0 & \text{caso contrário} \end{cases} \quad \forall i \neq j, \quad a_{ii} = 0, \quad \forall i.$$

A vantagem desta formulação inteira em relação à anteriormente apresentada reside no facto das restrições lineares estarem associadas aos nós do grafo, o que faz com que o número dessas restrições seja muito mais reduzido.

É possível demonstrar que a restrição de integralidade do problema (4.1) pode ser substituída por uma restrição linear. Com efeito verifica-se o seguinte resultado

Teorema 4.2 *Se G é um grafo, então*

$$\alpha(G) = \frac{1}{\min\{x^T(A + I)x : x \geq 0, e^T x = 1\}}$$

onde $e \in \mathbb{R}^n$ é um vector de componentes unitárias, A é a matriz de adjacência e I é a matriz identidade de ordem igual ao número de nós de G .

Demonstração: De [Motzkin e Straus, 1965], tem-se

$$\frac{1}{2} \left(1 - \frac{1}{\omega(G)} \right) = \max_x \left\{ \frac{1}{2} x^T A_G \mid x : e^T x = 1, x \geq 0 \right\},$$

com A_G a matriz de adjacência do grafo G . Por outro lado, do teorema 4.1 (i), vem

$$\frac{1}{2} \left(1 - \frac{1}{\alpha(G)} \right) = \max_x \left\{ \frac{1}{2} x^T A_{\bar{G}} \mid x : e^T x = 1, x \geq 0 \right\},$$

com \bar{G} o grafo complementar de G . Mas, pela definição de matriz de adjacência,

$$A_{\bar{G}} + A_G = ee^T - I.$$

Donde $A_{\bar{G}} = ee^T - A_G - I$ e

$$\frac{1}{2} x^T A_{\bar{G}} x = \frac{1}{2} (e^T x)^2 - \frac{1}{2} x^T (A_G + I)x.$$

Portanto

$$\begin{aligned} \frac{1}{2} \left(1 - \frac{1}{\alpha(G)} \right) &= \max_x \left\{ \frac{1}{2} (e^T x)^2 - \frac{1}{2} x^T (A_G + I)x : e^T x = 1, x \geq 0 \right\} = \\ &= \max_x \left\{ \frac{1}{2} - \frac{1}{2} x^T (A_G + I)x : e^T x = 1, x \geq 0 \right\} = \frac{1}{2} + \max_x \left\{ -\frac{1}{2} x^T (A_G + I)x : e^T x = 1, x \geq 0 \right\} = \\ &= \frac{1}{2} - \min_x \left\{ \frac{1}{2} x^T (A_G + I)x : e^T x = 1, x \geq 0 \right\} \end{aligned}$$

ou seja,

$$\frac{1}{2} \cdot \frac{1}{\alpha(G)} = \min_x \left\{ \frac{1}{2} x^T (A_G + I)x : e^T x = 1, x \geq 0 \right\}$$

o que demonstra o resultado pretendido. \square

Este resultado mostra que o número de independência de um grafo G se pode calcular a partir da resolução do programa quadrático

$$\begin{aligned} &\text{Minimizar} \quad \frac{1}{2} x^T (A + I)x \\ &\text{sujeito a} \quad e^T x = 1 \\ &\quad \quad \quad x \geq 0, \end{aligned} \tag{4.2}$$

com A a matriz de adjacência de G e I a matriz identidade de ordem igual ao número de nós. Após a determinação do mínimo global do programa quadrático, o número de independência é igual a metade do inverso do valor óptimo da função quadrática. Notar que este processo indica que as variáveis x_i com valores positivos correspondem aos vértices de G que pertencem ao conjunto de independência máximo.

Uma outra formulação quadrática referida na literatura [Bomze, 1998] para a determinação do conjunto independente máximo de um grafo corresponde ao seguinte resultado

$$\alpha(G) = \max_x \{2e^T x - x^T(A + I)x : x \geq 0\},$$

onde A é a matriz de adjacência e I é a matriz identidade de ordem igual ao número de nós de G . Assim um segundo processo para a determinação do número de independência determina uma solução óptima do programa

$$\begin{aligned} \text{Minimizar} \quad & -e^T x + \frac{1}{2} x^T(A + I)x \\ \text{sujeito a} \quad & x \geq 0. \end{aligned} \tag{4.3}$$

O número de independência é igual ao dobro do simétrico do valor óptimo desse programa. É de notar que este processo também indica os vértices que constituem o conjunto independente máximo e que são exactamente os correspondentes às variáveis com valor unitário na solução óptima. A equivalência entre os programas (4.2) e (4.3) foi estabelecida em [Bomze, 1998]. Na próxima secção iremos provar essa equivalência a partir das formulações *MPEC* desses programas quadráticos.

4.3 Formulações *MPEC*

Consideremos o programa quadrático (4.2)

$$\begin{aligned} \text{Minimizar} \quad & \frac{1}{2} x^T(A + I)x \\ \text{sujeito a} \quad & e^T x = 1 \\ & x \geq 0. \end{aligned}$$

As condições de *Karush-Kuhn-Tucker* desse programa constituem o *GLCP*

$$\begin{aligned} (A + I)x &= ue + w \\ e^T x &= 1 \\ x &\geq 0, w \geq 0 \\ x^T w &= 0, \end{aligned} \tag{4.4}$$

com $u \in \mathbb{R}$ e $w \in \mathbb{R}^n$ os multiplicadores de Lagrange associados às restrições $e^T x = 1$ e $x \geq 0$ respectivamente. Mas para qualquer solução do *GLCP* (4.4) tem-se

$$\frac{1}{2\alpha(G)} = \frac{1}{2} x^T(A + I)x = \frac{1}{2} x^T(ue + w) = \frac{1}{2} ue^T x = \frac{u}{2} \geq 0.$$

Portanto o valor inverso do número de independência pode ser obtido a partir do *MPEC*

$$\begin{aligned} \text{Minimizar} \quad & u = \frac{1}{\alpha(G)} \\ \text{sujeito a} \quad & w = (A + I)x - ue \\ & e^T x = 1 \\ & x \geq 0, w \geq 0, u \geq 0 \\ & x^T w = 0. \end{aligned} \tag{4.5}$$

É de notar que este *GLCP* se pode resolver a partir de um ponto estacionário do programa quadrático (4.2). Contudo a matriz associada às variáveis x é não quadrada e portanto não é *PSD*. Portanto o algoritmo complementar de pesquisa em árvore não pode ser usado para a determinação do mínimo global do *MPEC* (4.5). Os dois restantes métodos, sequencial complementar e enumerativo paramétrico, poderão ter utilidade para a resolução desse problema. Na secção 4.6 iremos discutir as adaptações destes processos para esse fim.

Uma segunda formulação *MPEC* para o cálculo do número de independência é obtida a partir do programa quadrático com restrições de não negatividade (4.3)

$$\begin{aligned} \text{Minimizar} \quad & -e^T x + \frac{1}{2} x^T (A + I)x \\ \text{sujeito a} \quad & x \geq 0. \end{aligned}$$

As suas condições de *Karush-Kuhn-Tucker* constituem o *LCP*

$$\begin{aligned} w &= -e + (A + I)x \\ x, w &\geq 0 \\ x^T w &= 0. \end{aligned}$$

Para cada solução desse *LCP*, tem-se

$$\begin{aligned} -\frac{1}{2} \alpha(G) &= -e^T x + \frac{1}{2} x^T (A + I)x = -\frac{1}{2} e^T x + \frac{1}{2} x^T (-e + (A + I)x) = \\ &= -\frac{1}{2} e^T x + \frac{1}{2} x^T w = -\frac{1}{2} e^T x. \end{aligned}$$

Portanto a determinação do conjunto independente máximo é equivalente ao *MPEC*

$$\begin{aligned} \text{Minimizar} \quad & -e^T x \\ \text{sujeito a} \quad & w = -e + (A + I)x \\ & x, w \geq 0 \\ & x^T w = 0. \end{aligned} \tag{4.6}$$

É de notar que a matriz do *LCP* é estritamente copositiva e portanto, pelo teorema 1.1, o *LCP* tem solução. Como a matriz não é *PSD*, então este *MPEC* não pode ser resolvido pelo método complementar de pesquisa em árvore. Por outro lado, o método enumerativo paramétrico também não deve ser usado para resolver o *MPEC*, pois o programa linear relaxado é ilimitado inferiormente, como é provado a seguir.

Teorema 4.3 *O programa linear relaxado do MPEC (4.6) é ilimitado.*

Demonstração: Consideremos o programa linear relaxado

$$\begin{aligned} \text{Minimizar} \quad & -e^T x \\ \text{sujeito a} \quad & (A + I)x \geq e \\ & x \geq 0. \end{aligned}$$

Tal como na prova do teorema 1.5, para estabelecer o resultado pretendido é necessário existir um vector x não negativo e não nulo satisfazendo

$$(A + I)x \geq 0, \quad -e^T x < 0.$$

Como A é uma matriz não negativa e I é a matriz identidade, então qualquer vector x não negativo e não nulo satisfaz essa condição. \square

Seguidamente estabelecemos que os *MPECs* (4.5) e (4.6) são equivalentes, o que implica imediatamente a equivalência entre os programas quadráticos (4.2) e (4.3) referida na secção anterior.

Teorema 4.4 *Os MPECs (4.5) e (4.6) são equivalentes.*

Demonstração: (i) Se (\bar{x}, \bar{w}) é solução óptima do *MPEC* (4.6), então $\bar{x} \neq 0$ e portanto $e^T \bar{x} > 0$. Mas

$$\bar{w} = -e + (A + I)\bar{x} \Leftrightarrow \frac{\bar{w}}{e^T \bar{x}} = -\frac{e}{e^T \bar{x}} + (A + I) \left(\frac{\bar{x}}{e^T \bar{x}} \right).$$

Se $x = \frac{\bar{x}}{e^T \bar{x}}$, então $e^T x = 1$. Além disso, se escrevermos $u = \frac{1}{e^T \bar{x}}$ e $w = \frac{\bar{w}}{e^T \bar{x}}$, vem

$$w = -ue + (A + I)x$$

$$e^T x = 1.$$

Como os conjuntos admissíveis dos *GLCPs* (4.5) e (4.6) são equivalentes e $u = \frac{1}{e^T \bar{x}}$ então (x, w, y) é solução óptima do *MPEC* (4.5).

(ii) Se $(\bar{x}, \bar{w}, \bar{u})$ é solução óptima do *MPEC* (4.5), então $0 \neq \bar{x} \geq 0$. Por outro lado

$$0 = \bar{x}^T \bar{w} = \bar{x}^T (A + I)\bar{x} - \bar{u}(e^T \bar{x}) \Rightarrow \bar{u} = \bar{x}^T (A + I)\bar{x} > 0.$$

Donde

$$x = \frac{\bar{x}}{\bar{u}}, \quad w = \frac{\bar{w}}{\bar{u}}$$

é solução admissível do *GLCP* (4.6). Além disso

$$e^T x = \frac{e^T \bar{x}}{\bar{u}} = \frac{1}{\bar{u}}.$$

Então, como anteriormente, (x, w) é solução óptima do *MPEC* (4.6). \square

Como última formulação *MPEC* para a determinação do conjunto independente máximo, consideremos o programa de dois níveis do problema, apresentado em [Frangioni, 1995]

$$\begin{aligned}
 & \text{Maximizar} && e^T y \\
 & \text{sujeito a} && x_i + x_j \leq 1, \forall (i, j) \in E \\
 & && 0 \leq x \leq e \\
 & && y \text{ é solução óptima de} \\
 & && \max_y (x - e)^T y \\
 & && \text{sujeito a } 0 \leq y \leq e.
 \end{aligned} \tag{4.7}$$

É de notar que da definição do problema de dois níveis se tem

$$0 \leq x_i < 1 \Rightarrow y_i = 0$$

para todo o $i = 1, \dots, n$. Por outro lado se $x_i = 1$, então y_i pode tomar um qualquer valor. A maximização da função linear $e^T y$ no primeiro nível faz com que $y_i = 1$ para todo $x_i = 1$. Essas variáveis positivas correspondem aos vértices do conjunto independente máximo. O dual do programa linear do segundo nível nas variáveis y_i é

$$\begin{aligned}
 & \text{Minimizar} && e^T u \\
 & \text{sujeito a} && u \geq x - e \\
 & && u \geq 0.
 \end{aligned}$$

De acordo com a teoria da dualidade linear [Murty, 1983], o programa do segundo nível pode ser substituído por um problema linear complementar correspondente às admissibilidades primal e dual e à restrição de complementaridade das variáveis de folga. Se introduzirmos as variáveis de folga w e v para as restrições $y \leq e$ e $u \geq x - e$ respectivamente, podemos escrever o programa de dois níveis (4.7) como o seguinte *MPEC*

$$\begin{aligned}
 & \text{Minimizar} && -e^T y \\
 & \text{sujeito a} && x_i + x_j \leq 1, \forall (i, j) \in E \\
 & && 0 \leq x \leq e \\
 & && w = e - y \\
 & && v - u = e - x \\
 & && w, y, v, u \geq 0 \\
 & && u^T w = v^T y = 0.
 \end{aligned} \tag{4.8}$$

Seja agora (w, y, v, u, x) uma solução do *GLCP* correspondente às restrições do *MPEC*. Então

$$(i) \quad v_i > 0 \Rightarrow y_i = 0 \Rightarrow w_i = 1 \Rightarrow u_i = 0.$$

$$(ii) \quad u_i > 0 \Rightarrow w_i = 0 \Rightarrow y_i = 1 \Rightarrow v_i = 0.$$

(iii) $x_i = 1 - v_i + u_i$ e $x_i \leq 1 \Rightarrow 1 - v_i + u_i \leq 1 \Rightarrow u_i \leq v_i$.

Portanto $u_i = 0$ para todo o $i = 1, \dots, n$ e a restrição $u^T w = 0$ é redundante. Mas

$$\begin{aligned} x_i = 1 - v_i \geq 0 &\Leftrightarrow v_i \leq 1 \\ v_i = 1 - x_i \leq 1 &\Leftrightarrow v_i \geq 0 \\ x_i + x_j \leq 1 &\Leftrightarrow 1 - v_i + 1 - v_j \leq 1 \Leftrightarrow v_i + v_j \geq 1. \end{aligned}$$

Podemos assim suprimir as variáveis x_i do programa (4.8) e obter a seguinte formulação *MPEC* equivalente

$$\begin{aligned} \text{Minimizar} \quad & -e^T y \\ \text{sujeito a} \quad & v_i + v_j \geq 1, \forall (i, j) \in E \\ & 0 \leq v \leq e \\ & 0 \leq y \leq e \\ & v^T y = 0. \end{aligned} \tag{4.9}$$

Uma vez resolvido este *MPEC*, então o conjunto independente máximo é constituído pelos vértices associados às variáveis $y_i = 1$ na solução óptima obtida. É de notar que a matriz correspondente às variáveis v não é *PSD*, pelo que o algoritmo complementar de pesquisa em árvore também não tem utilidade para este *MPEC*. Na secção 4.6 iremos discutir a resolução deste *MPEC* pelas duas restantes técnicas alternativas para a determinação de um mínimo global do *MPEC*. É ainda de acrescentar que, contrariamente às duas formulações anteriormente apresentadas nesta secção, o número de restrições que não são limites inferiores ou superiores é igual ao número de arestas do grafo. Essa é a grande desvantagem desta formulação, pois o número de arestas de um grafo é em geral muito superior ao seu número de nós. As vantagens desta formulação residem nos factos da função objectivo ser linear e do programa relaxado ter solução óptima.

4.4 Algoritmo de Diferenças Convexas

Nesta secção discutimos um algoritmo de ramificação e limitação para a resolução do programa quadrático (4.2)

$$\begin{aligned} \text{Minimizar} \quad & f(x) = \frac{1}{2}x^T(A + I)x \\ \text{sujeito a} \quad & e^T x = 1 \\ & x \geq 0. \end{aligned} \tag{4.10}$$

Este processo é baseado nas ideias apresentadas em [Horst et al., 2000], em que a habitual ramificação em árvore é visualizada pela partição do conjunto admissível relaxado em conjuntos mais pequenos, nos quais são determinados limites inferiores e superiores e a ligação entre dois nodos representa a obtenção do segundo conjunto por partição directa do primeiro. Introduzem-se de seguida algumas noções topológicas relevantes para a descrição do processo.

Definição 4.3 *Um simplex de dimensão k ou um k -simplex é um conjunto das combinações convexas dos vértices $\{v_1, v_2, \dots, v_{k+1}\}$.*

Da definição anterior resulta que um vértice é um 0-simplex, um segmento de recta é um 1-simplex e um triângulo é um 2-simplex. Deste modo o conjunto admissível do problema quadrático definido por

$$S = \{x \in \mathbb{R}^n : e^T x = 1, x \geq 0\}$$

é o $(n - 1)$ -simplex com os vértices

$$e^i = [0 \dots 0 \underbrace{1}_i 0 \dots 0]^T, \quad i \in \{1, \dots, n\}.$$

Na ramificação em árvore o algoritmo utiliza a noção de subdivisão radial de um simplex S , que se expõe seguidamente.

Definição 4.4 *Seja S um $(n - 1)$ -simplex com vértices $V(S) = \{v_1, \dots, v_n\}$. Seja $w \in S$, $w \notin V(S)$ representado univocamente por*

$$w = \sum_{i=1}^n \lambda_i v_i,$$

com $\lambda_i \geq 0$, $i = 1, \dots, n$ e

$$\sum_{i=1}^n \lambda_i = 1.$$

Para cada $i = 1, \dots, n$ tal que $\lambda_i > 0$, considere o simplex

$$S(i, w) = \text{conv}\{v_1, \dots, v_{i-1}, w, v_{i+1}, \dots, v_n\}.$$

Esta partição denomina-se subdivisão radial.

É possível demonstrar [Horst e Tuy, 1996] que cada $S(i, w)$ é um $(n - 1)$ -simplex e que

$$S = \bigcup_{i: \lambda_i > 0} S(i, w), \quad \text{int } S(i, w) \cap \text{int } S(j, w) = \emptyset, \quad i \neq j,$$

com $\text{int } A$ o interior com respeito ao espaço de dimensão $(n - 1)$ de S .

Definição 4.5 *Designa-se por bissecção, a subdivisão radial em que w é escolhido por forma a ser o ponto médio de uma das maiores arestas de S .*

Em cada iteração do algoritmo utiliza-se um conjunto Q de pontos admissíveis que é sucessivamente actualizado. O conjunto inicial é constituído pelos vértices $e^i, i = 1, \dots, n$ e pelo baricentro

$$b = \frac{1}{n} \sum_{i=1}^n e^i = \frac{1}{n} [1 \ 1 \ \dots \ 1]^T \in \mathbb{R}^n.$$

O simplex inicial é então dividido em n simplices S_i de igual dimensão. Para formar S_i , $i = 1, \dots, n$ substitui-se o vértice original e^i pelo baricentro b . Assim obtém-se

$$S_i = \text{conv}\{p_1, \dots, p_{i-1}, w, p_{i+1}, \dots, p_n\}, \quad i = 1, \dots, n,$$

com $p_j = e^j$ e $w = b$. A partir da segunda iteração é utilizada uma bissecção, ou seja, w é escolhido como sendo o ponto médio de uma das maiores arestas do simplex S actual, obtendo-se assim dois novos simplices com o mesmo volume. Para as sucessivas bissecções verifica-se a seguinte propriedade.

Lema 4.1 *Para cada sucessão $\{S_q\}$, $S_{q+1} \subset S_q$ de sucessivas subdivisões do simplex por bissecção, existe um ponto s tal que $\lim_{q \rightarrow \infty} S_q = \bigcap_q S_q = \{s\}$, ou seja, a sucessão S_q eventualmente reduz-se a um ponto.*

A demonstração deste lema pode ser consultada por exemplo em [Horst et al., 2000]. Seguidamente apresentamos os passos do algoritmo, para depois nos debruçarmos sobre a sua convergência.

ALGORITMO DE DIFERENÇAS CONVEXAS

Passo Inicial

Seja $Q = \{e^1, e^2, \dots, e^n\}$, $\gamma \leftarrow \min\{f(x) : x \in Q\}$ e escolha o ponto $v \in Q$ tal que $f(v) = \gamma$. Determine um limite inferior $\mu(s)$ para $f(x)$ em S . Faça $\mu \leftarrow \mu(s)$, $\tilde{S} \leftarrow \{S\}$ e $k = 1$.

Passo Geral (Iteração k)

Se $\gamma = \mu$ termine com v a solução óptima e γ o valor óptimo do problema (4.10). Caso contrário faça a subdivisão de S em simplices, S_1, \dots, S_r ($r = n$ inicialmente e $r = 2$ após a primeira iteração). Para $i = 1, \dots, r$ determine um limite inferior $\mu(S_i)$ para f em S_i , que satisfaz $\mu(S_i) \geq \mu$ e calcule o conjunto finito $Q(S_i) \subset S_i$. Faça

$$Q \leftarrow Q \cup \{Q(S_i) : i = 1, \dots, r\}.$$

$$\gamma \leftarrow \min\{f(x) : x \in Q\} \text{ e escolha } v \in Q \text{ tal que } f(v) = \gamma.$$

$$\tilde{S} \leftarrow \tilde{S} \setminus \{S\} \cup \{S_i : \mu(S_i) < \gamma, i = 1, \dots, r\}.$$

$$\mu \leftarrow \begin{cases} \gamma & \text{se } \tilde{S} = \emptyset \\ \min\{\mu(S) : S \in \tilde{S}\} & \text{caso contrário.} \end{cases}$$

Escolha $S \in \tilde{S}$ tal que $\mu(S) = \mu$, faça $k \leftarrow k + 1$ e repita este Passo.

Para uma descrição completa deste processo, é necessário explicar a forma como se calculam os limites inferiores do Passo Geral. Uma das abordagens mais usadas em optimização global consiste em escrever a função objectivo na diferença de duas funções convexas g e h . A substituição da segunda função h por uma função linear $c^T x$ tal que

$$\forall x \in S \quad c^T x \geq h(x),$$

permite determinar uma função convexa t em S , que satisfaz

$$\forall x \in S \quad f(x) \geq g(x) - c^T x = t(x).$$

Como a função t é convexa em S , então um seu mínimo global pode ser determinado a partir de um ponto estacionário do programa

$$\begin{aligned} & \text{Minimizar} \quad t(x) = g(x) - c^T x \\ & \text{sujeito a} \quad e^T x = 1 \\ & \quad \quad \quad x \geq 0. \end{aligned}$$

O valor óptimo desse programa fornece um limite inferior pretendido. A dificuldade deste processo está na construção da função linear $c^T x$. Seguidamente discutimos um processo para determinar esse limite inferior que não usa a expressão explícita da função linear. É perfeitamente conhecido que a matriz simétrica $(A + I)$ se pode escrever na forma

$$A + I = A^+ - A^-, \quad (4.11)$$

com A^+ e A^- matrizes de ordem n semi-definidas positivas. Existem várias maneiras de conseguir esta decomposição, que serão discutidas mais adiante.

Seja S o simplex corrente para o qual se pretende calcular o limite inferior $\mu(S)$ e utilize-se a equação (4.11) para obter o programa

$$\begin{aligned} & \text{Minimizar} \quad \frac{1}{2} x^T A^+ x - \frac{1}{2} x^T A^- x \\ & \text{sujeito a} \quad x \in S. \end{aligned}$$

O limite superior da função quadrática convexa $\frac{1}{2} x^T A^- x = h(x)$ pode ser obtido através da estimação convexa $F(x)$ em S , normalmente designada por invólucro convexo, que aparece discutido em [Horst et al., 2000].

Lema 4.2 *O invólucro convexo de uma função convexa h , real de variável real, sobre um simplex S é a função afim que coincide com h nos vértices de S .*

Consideremos agora o simplex e os vértices v_1, \dots, v_n de S . Então cada $x \in S$ pode ser univocamente determinado por

$$x = \sum_{i=1}^n \alpha_i v_i,$$

com $\alpha_i \geq 0, i = 1, \dots, n$ e $\sum_{i=1}^n \alpha_i = 1$. Como é mostrado em [Horst et al., 2000], o invólucro convexo é obtido pela expressão

$$F(x) = \frac{1}{2} \sum_{i=1}^n \alpha_i v_i^T A^- v_i.$$

Desta forma um limite inferior pode ser obtido através da resolução do seguinte programa quadrático convexo

$$\begin{aligned} \text{Minimizar} \quad & \frac{1}{2} \left(\sum_{i=1}^n \alpha_i v_i \right)^T A^+ \left(\sum_{i=1}^n \alpha_i v_i \right) - \frac{1}{2} \sum_{i=1}^n \alpha_i v_i^T A^- v_i \\ \text{sujeito a} \quad & \sum_{i=1}^n \alpha_i = 1 \\ & \alpha_i \geq 0, i = 1, \dots, n. \end{aligned} \tag{4.12}$$

Para cada simplex S é possível obter um outro limite inferior $\mu_2(S)$ pior do que o obtido a partir da solução ótima do programa (4.12). Para a sua determinação, notemos que, como $\frac{1}{2}x^T A^+ x$ é uma função convexa, então

$$\frac{1}{2}x^T A^+ x \geq \frac{1}{2}x_0^T A^+ x_0 + (x - x_0)^T A^+ x_0 = g(x) \tag{4.13}$$

para qualquer $x_0 \in S$ dado. Por exemplo, x_0 pode ser o baricentro de S , isto é,

$$x_0 = \frac{1}{n} \sum_{i=1}^n v_i$$

com v_i os vértices do simplex S . Se substituirmos a função quadrática $\frac{1}{2}x^T A^+ x$ pela função linear $g(x)$, dada por (4.13), obtemos um programa linear em S . Como a solução ótima desse programa é um dos vértices $v_t, t \in \{1, 2, \dots, n\}$, então é fácil de concluir que o seu valor ótimo satisfaz

$$\mu_2(S) = g(v_t) \leq \mu_1(S). \tag{4.14}$$

Portanto o limite inferior $\mu_2(S)$ não deve ser usado na prática, mas é importante para o estabelecimento da convergência do algoritmo, que é apresentada a seguir.

Teorema 4.5 *Se o algoritmo de diferenças convexas tem um número infinito de iterações, então*

$$\mu = \lim_{k \rightarrow \infty} \mu_k = \lim_{k \rightarrow \infty} f(v_k) = \lim_{k \rightarrow \infty} \gamma_k = \gamma,$$

com μ_k e γ_k os limites inferior e superior em cada iteração k respectivamente e cada ponto de acumulação v^* da sucessão $\{v_k\}$ é uma solução ótima do programa (4.10).

Demonstração: Se o algoritmo é infinito, então deve gerar pelo menos uma sucessão infinita $\{S_q\}$ de simplices sucessivos satisfazendo $\{S_q\} \supset \{S_{q+1}\}$. Vamos primeiramente mostrar que em cada uma destas sucessões se verifica

$$\lim_{q \rightarrow \infty} (\gamma_q - \mu_q) = \lim_{q \rightarrow \infty} (\gamma_q - \mu(S_q)) = 0, \quad (4.15)$$

com S_q o simplex usado no cálculo do limite inferior. Notemos que basta considerar o limite inferior μ_2 , pois o primeiro limite μ_1 é pelo menos tão bom como μ_2 .

Seja $F_{S_q}(x)$ a função objectivo do problema correspondente ao segundo limite inferior e seja $x^*(S_q)$ a solução óptima deste problema. Dada a linearidade de $F_{S_q}(x)$, $x^*(S_q)$ é um vértice óptimo de S_q . Iremos seguidamente mostrar que

$$\lim_{q \rightarrow \infty} [f(x^*(S_q)) - F_{S_q}(x^*(S_q))] = 0. \quad (4.16)$$

Este resultado implica (4.15), pois

$$\mu(S_q) = F_{S_q}(x^*(S_q)) \leq \gamma_q \leq f(x^*(S_q)).$$

Pelo lema 4.1 tem-se

$$\lim_{q \rightarrow \infty} S_q = \{s\} \quad \text{e} \quad \lim_{q \rightarrow \infty} x_q = s,$$

para qualquer sucessão $\{x_q\}$, $x_q \in S_q$, $q = 1, 2, \dots$. A continuidade de f implica

$$\lim_{q \rightarrow \infty} f(x^*(S_q)) = f(s).$$

Como estamos a considerar o limite inferior μ_2 , a função $F_{S_q}(x)$ é da forma

$$l_{S_q}^1(x) - l_{S_q}^2(x)$$

com $l_{S_q}^1(x)$ e $l_{S_q}^2(x)$ funções lineares. A função $l_{S_q}^1(x)$ resulta de substituir a parte convexa $\frac{1}{2}x^T A^+ x$ pelo seu hiperplano tangente (aproximação de Taylor de primeira ordem) no baricentro x_{0,S_q} de S_q . Dado que

$$\lim_{q \rightarrow \infty} x_{0,S_q} = s$$

obtemos novamente pela continuidade das funções envolvidas,

$$\lim_{q \rightarrow \infty} l_{S_q}^1(x^*(S_q)) = \frac{1}{2}s^T A^+ s.$$

A função $l_{S_q}^2(x)$ é a função linear que coincide com $\frac{1}{2}x^T A^- x$ nos vértices de S_q . Analogamente, tem-se

$$\lim_{q \rightarrow \infty} l_{S_q}^2(x^*(S_q)) = \frac{1}{2}s^T A^- s.$$

Tendo em conta a decomposição da função objectivo f do programa quadrático dado, vem

$$\lim_{q \rightarrow \infty} F_{S_q}(x^*(S_q)) = \lim_{q \rightarrow \infty} f(x^*(S_q)) = f(s)$$

e portanto verificam-se (4.16) e (4.15).

Para provar o teorema, seja $\{v_k\}$ uma sucessão de vértices de simplices gerados pelo algoritmo. Como esta sucessão é limitada, então admite um ponto de acumulação v^* e existe uma subsucessão infinita de $\{v_k\}$ convergente para v^* . Esta sucessão tem de conter uma subsucessão infinita, que denotamos por $\{v_q\}$, para a qual a correspondente sucessão $\{S_q\}$ satisfaz $\{S_q\} \supset \{S_{q+1}\}$, $q = 1, 2, \dots$.

Seja f^* o valor óptimo da função associada ao cálculo do segundo limite inferior. A sucessão $\{\mu_k\}$ de limites inferiores é obviamente não decrescente e limitada superiormente por f^* , pelo que é convergente. Seja

$$\mu = \lim_{k \rightarrow \infty} \mu_k.$$

Analogamente a sucessão dos limites superiores é não crescente e limitada inferiormente, pois

$$\gamma_k = f(v_k) \geq f^*.$$

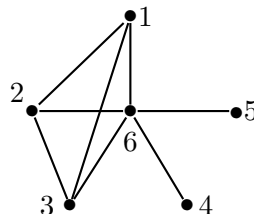
Então é convergente para γ , que satisfaz

$$\mu \leq f^* \leq \lim_{k \rightarrow \infty} f(v_k) = f(v^*) = \gamma. \quad (4.17)$$

O resultado é agora por consequência de (4.17) e (4.15). \square

Na prática, é conveniente determinar um limite superior inicial a partir de um ponto estacionário do programa original. Além disso, como referimos anteriormente, os limites inferiores são obtidos a partir da determinação de pontos estacionários (mínimos globais) dos programas quadráticos da forma (4.12).

Seguidamente apresentamos um exemplo do processo de resolução que acabámos de descrever. Antes de prosseguirmos, convém salientar que o estudo é efectuado sobre o grafo complementar do grafo G , pois para todos os exemplos testados na secção 4.7 os valores óptimos conhecidos são os da clique máxima. Consideremos então o grafo G introduzido na secção 4.1



A matriz $(A_{\bar{G}} + I)$ tem a seguinte forma

$$(A_{\bar{G}} + I) = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Uma forma de construir A^+ é tornar A diagonalmente dominante estrita com elementos diagonais positivos [Cottle et al., 1992] por acréscimo de números positivos à diagonal. Por exemplo, podemos escolher A^+ a partir de

$$A^+ = \begin{bmatrix} 3 & 0 & 0 & 1 & 1 & 0 \\ 0 & 3 & 0 & 1 & 1 & 0 \\ 0 & 0 & 3 & 1 & 1 & 0 \\ 1 & 1 & 1 & 5 & 1 & 0 \\ 1 & 1 & 1 & 1 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Dado que $(A_{\bar{G}} + I) = A^+ - A^-$, obtém-se a matriz A^- definida por

$$A^- = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

O algoritmo começa por determinar no nó inicial os valores 0.1250 e -0.7500 para os limites superior e inferior, respectivamente. De acordo com a exposição do algoritmo, são calculados seguidamente os valores dos limites inferiores nos seis nós de nível 1, que se apresentam na tabela seguinte

Nós	1	2	3	4	5	6
LIMITE INFERIOR	-0.7083	-0.7083	-0.7083	-0.6250	-0.6250	-0.7500

Para gerar os nós do próximo nível, o algoritmo selecciona primeiramente o nó 6, associado ao menor limite inferior, que origina os dois nós a seguir apresentados

Nós	7	8
LIMITE INFERIOR	-0.7000	-0.7000

e elimina o nó 6 da lista. Seguidamente o nó 2 é seleccionado da lista, por apresentar o menor limite inferior, sendo esse nó substituído por dois novos nós com valores

Nós	9	10
LIMITE INFERIOR	-0.6667	-0.6667

O processo é repetido e vão sendo gerados sucessivos nós de acordo com as regras do processo. Constata-se que a alteração nos valores dos limites inferiores é mínima, mostrando a lentidão do método na obtenção da solução óptima. Com efeito, o algoritmo necessita de gerar 703 nós e de efectuar 5812 operações pivotais para determinar a solução óptima do programa dado e assim calcular o conjunto independente máximo do grafo. Outras escolhas para a matriz A^+ podem ser feitas com vantagem. Assim, se $\lambda_{\min}(A)$ representa o menor valor próprio de $(A + I)$, então uma possível escolha para A^+ é

$$A_1^+ = A + (1 - \lambda_{\min}(A))I$$

com I a matriz identidade. Por outro lado, a função quadrática $x^T A x$ é convexa em

$$K = \{x \in \mathbb{R}^n : e^T x = 1\}$$

se e só se $Z^T(A + I)Z \in PSD$, para qualquer matriz Z contendo vectores de uma base do espaço nulo de e^T . Por isso uma outra escolha é

$$A_2^+ = A + (1 - \lambda_{\min}(Z^T A Z))I.$$

Além disso, como a matriz $(A + I)$ é diagonal por blocos e o segundo bloco diagonal [1] é PD , então uma outra escolha para A^+ consiste em modificar apenas o bloco diagonal que não é PD . Se usarmos o menor valor próprio de A , obtemos a matriz

$$A_3^+ = \begin{bmatrix} 2 & 0 & 0 & 1 & 1 & 0 \\ 0 & 2 & 0 & 1 & 1 & 0 \\ 0 & 0 & 2 & 1 & 1 & 0 \\ 1 & 1 & 1 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Na próxima tabela incluímos os resultados da aplicação do algoritmo com estas três escolhas para a matriz A^+ . Nessa tabela, IT representa o número de pivotagens e T o tempo em segundos para a determinação do conjunto independente máximo.

MATRIZ	NÓS	IT	T
A_1^+	215	1824	2.47
A_2^+	179	1539	1.77
A_3^+	107	1065	1.20

Facilmente se verifica a enorme dependência do comportamento do algoritmo em relação à escolha da matriz A^+ da decomposição de A numa diferença de duas matrizes PSD .

Além disso, mesmo no caso da melhor escolha, o algoritmo necessita de um grande esforço computacional para determinar uma solução ótima do programa quadrático. Experiência computacional com este algoritmo em problemas de maiores dimensões indicou que o processo não é muito eficaz para a determinação do conjunto independente máximo, pelo que novas técnicas heurísticas serão necessárias de modo a torná-lo competitivo com outros processos que exploram as formulações *MPEC* e que são discutidos a seguir.

4.5 Determinação de um Conjunto Independente

Na secção 4.3 mostrámos que a determinação do conjunto de independência máximo se reduz à resolução de um *MPEC* por intermédio de uma de três formas diferentes. A primeira formulação (4.5) surgiu associada ao programa quadrático (4.2) e o *GLCP* corresponde às suas condições de *Karush-Kuhn-Tucker*. Por isso uma primeira solução do *GLCP* do *MPEC* (4.5) deve ser calculada a partir da determinação de um ponto estacionário de (4.2). Desse modo obtém-se um conjunto independente, não necessariamente máximo, correspondente às variáveis x_i positivas da solução do *GLCP*.

O segundo *MPEC* (4.6) apresentado na secção 4.3 tem como conjunto de restrições um *LCP* com uma matriz estritamente copositiva. É perfeitamente conhecido que esse tipo de *LCP* tem sempre solução, que se pode obter através do Método de *Lemke* [Murty, 1988, Cottle et al., 1992]. Como anteriormente, essa solução fornece um conjunto independente correspondente à variáveis x_i positivas (e iguais a um).

Seguidamente apresentamos uma descrição sumária deste algoritmo, por muitos considerado o mais popular para a resolução do *LCP* [Murty, 1988, Cottle et al., 1992]. Nesse processo é introduzida uma variável adicional $z_0 \geq 0$ e um vector $d \geq 0$ tal que $q + \bar{z}_0 d \geq 0$ para certo valor $\bar{z}_0 > 0$ dessa variável. Desse modo obtém-se um *GLCP*

$$\begin{aligned} w &= q + z_0 d + Mz \\ z &\geq 0, w \geq 0, z_0 \geq 0 \\ z^T w &= 0. \end{aligned}$$

É evidente que uma solução deste *GLCP* é solução do *LCP* se $z_0 = 0$. O método de *Lemke* procura obter uma tal solução do *GLCP* por uma pesquisa inteligente das suas soluções. O algoritmo inicia-se com uma solução básica do *GLCP* tal que $z_0 > 0$ seja básica e para a qual existe um só par de variáveis complementares (z_i, w_i) não básicas. Ao longo do processo são sempre usadas esse tipo de soluções básicas até à obtenção de uma solução do *GLCP* com $z_0 = 0$ ou uma indicação de que tal não é possível de fazer de acordo com as regras do processo. Para manter este tipo de soluções básicas, a coluna pivotal é escolhida como sendo a da variável complementar da que passou a não básica na iteração anterior e a linha pivot é calculada usando o critério do quociente mínimo semelhante ao do método do simplex para programas lineares. Deste modo o processo termina quando se encontra uma solução com

$z_0 = 0$ ou numa aresta ilimitada se a linha pivotal não puder ser escolhida. Os passos do algoritmo são apresentados a seguir.

ALGORITMO DE LEMKE

Considere-se o conjunto de restrições do *GLCP* definido pelo sistema

$$\begin{aligned} Ax &= q \\ x &\geq 0 \end{aligned}$$

com

$$A = [I \ -M \ -d] \in \mathbb{R}^{n \times (2n+1)}$$

e

$$x^T = [w \quad z \quad z_0] \in \mathbb{R}^{2n+1}.$$

Passo Inicial

Se $q \geq 0$ então $w = q$ e $z = 0$ é solução do *LCP*. Caso contrário, determine r a partir de

$$r = \min \left\{ s : \frac{q_s}{d_s} = \min \left\{ \frac{q_i}{d_i} : q_i < 0 \right\} \right\}.$$

Efectue uma operação pivotal com pivot d_r . A variável $x_{2n+1} = z_0$ passa a básica por troca com a variável básica $x_r = w_r$.

Passo Geral

Seja u_r a variável complementar da que passou a não básica na iteração anterior e $\bar{A} = [\bar{a}_{ij}]$ e $\bar{q} = [\bar{q}_i]$ a matriz e o vector dos valores das variáveis básicas associadas à solução básica corrente. Se

$$\bar{a}_{ir} \leq 0 \quad \forall i = 1, \dots, n$$

o método termina em aresta ilimitada. Caso contrário calcule

$$t = \min \left\{ s : \frac{\bar{q}_s}{\bar{a}_{sr}} = \min \left\{ \frac{\bar{q}_i}{\bar{a}_{ir}} : \bar{a}_{ir} > 0 \right\} \right\}. \quad (4.18)$$

Efectue uma operação pivotal com pivot \bar{a}_{tr} . Se $z_0 = 0$ obtém-se uma solução do *LCP*. De outro modo repita este Passo.

Este algoritmo tem convergência finita, desde que as soluções básicas utilizadas durante a resolução do *LCP* sejam todas não degeneradas [Murty, 1988]. A terminação em aresta ilimitada não significa obrigatoriamente que o *LCP* não tenha solução. Daí que a sua aplicabilidade seja restrita a *LCPs* cuja matriz M pertence a determinadas classes. Em particular, é possível mostrar [Murty, 1988, Cottle et al., 1992] que o algoritmo obtém uma solução do *LCP* se M é estritamente copositiva. Portanto o algoritmo é capaz de obter

sempre um conjunto independente para o problema. É ainda de notar que a regra de *Bland* [Murty, 1988] é usada na escolha da linha pivotal em cada iteração, de modo a lidar com soluções degeneradas.

De seguida apresentamos o exemplo dos passos do algoritmo na determinação de um conjunto independente para o grafo introduzido na secção 4.1. Utilizamos o vector d usual de coordenadas unitárias, ou seja, $d = [1, \dots, 1]^T$. No passo inicial do algoritmo existe um empate quanto à variável que deve abandonar a base, uma vez que $\frac{q_i}{d_i} = -1, \forall i = 1, \dots, n$. De acordo com a regra de *Bland* (4.18), w_1 é seleccionada para sair da base por troca com a variável z_0 , obtendo-se uma solução com variáveis básicas dadas na tabela a seguir

BASE	z_0	w_2	w_3	w_4	w_5	w_6
VALOR DAS BÁSICAS	1	0	0	0	0	0

Na segunda iteração a variável z_1 , complementar de w_1 , entra na base por troca com w_2 , que é uma vez mais determinada a partir da regra de *Bland*. Desse modo obtém-se a solução básica com as seguintes variáveis básicas

BASE	z_0	z_1	w_3	w_4	w_5	w_6
VALOR DAS BÁSICAS	1	0	0	0	0	0

Na terceira iteração z_2 troca com w_3 e obtém-se a solução

BASE	z_0	z_1	z_2	w_4	w_5	w_6
VALOR DAS BÁSICAS	1	0	0	0	0	0

Na quarta iteração z_3 troca com w_6 e tem-se

BASE	z_0	z_1	z_2	w_4	w_5	z_3
VALOR DAS BÁSICAS	1	0	0	0	0	0

Finalmente na quinta iteração z_6 troca com z_0 e obtém-se uma solução complementar, cujos valores das variáveis básicas são apresentados a seguir.

BASE	z_6	z_1	z_2	w_4	w_5	z_3
VALOR DAS BÁSICAS	1	1	1	2	2	1

O conjunto independente $\{1, 2, 3, 6\}$ obtido é o de máxima cardinalidade e é constituído pelos nós correspondentes às variáveis z_i básicas. Da resolução do problema verifica-se que as soluções básicas obtidas em cada iteração do método de *Lemke* são altamente degeneradas. Como a regra de *Bland* é usada nos casos degenerados, então o algoritmo vai depender da forma como se dispõem as linhas e colunas da matriz A antes do problema ser resolvido pelo

método de *Lemke*. Esse assunto será discutido mais adiante.

4.6 Algoritmo Sequencial Complementar

Consideremos a formulação *MPEC* (4.5) do problema da determinação do conjunto de independência máximo. Como vimos na secção anterior, um método local de programação quadrática permite determinar um conjunto independente. Seja \bar{u} o valor da função objectivo desse *MPEC* no ponto estacionário. Então os próximos *GLCP*(λ_k), $k = 1, 2, \dots$, a resolver têm a forma

$$\begin{aligned} w &= (A + I)x - ue \\ e^T x &= 1 \\ x &\geq 0, w \geq 0 \\ 0 &\leq u \leq \lambda_k \\ x^T w &= 0 \end{aligned} \tag{4.19}$$

onde

$$\lambda_k = \lambda_{k-1} - \gamma \lambda_{k-1}, \quad k = 1, 2, \dots, \quad \lambda_0 = \bar{u} \tag{4.20}$$

com $\gamma > 0$ uma tolerância para zero. O método sequencial complementar vai resolvendo esses *GLCPs* e há dois casos:

- (i) *GLCP*(λ_k) não tem solução e a solução do *GLCP*(λ_{k-1}) fornece uma solução para o conjunto de independência máximo.
- (ii) *GLCP*(λ_k) tem uma solução e λ_k é actualizado a partir de (4.20) e um novo *GLCP* é resolvido.

Como todos os *GLCP*(λ_k) a partir do primeiro não têm uma estrutura especial, então o método enumerativo deve ser sempre usado para os resolver. De notar que neste caso o *GLCP* é um problema de optimização paramétrico com um único parâmetro, pelo que técnicas de optimização complementar paramétrica poderão ser úteis para a sua resolução.

A aplicação do algoritmo sequencial complementar para a resolução dos dois restantes *MPECs* não apresenta qualquer novidade em relação ao método usual. Dado que o método de *Lemke* determina eficientemente um conjunto independente, então deve-se usar inicialmente $\lambda_0 = -p$, com p a cardinalidade do conjunto obtido por esse processo.

A utilidade do método enumerativo paramétrico para a resolução destes *MPECs* não parece ser muito interessante, pois os respectivos programas lineares relaxados ou não têm solução óptima ou têm sempre uma solução óptima trivial ao longo do processo. Por isso não será usado nas nossas experiências.

4.7 Experiência Computacional

Os problemas teste utilizados na determinação do *MIS* foram retirados da colecção *Dimacs* [Johnson e Trick, 1996], que engloba uma vasta gama de grafos. As características dos problemas utilizados apresentam-se na Tabela 4.1. É de referir que os valores óptimos conhecidos para estes problemas são os valores da clique máxima do grafo G , pelo que o estudo para a determinação do *MIS* foi efectuado sobre o grafo \bar{G} .

Tabela 4.1: *Características dos problemas teste para determinação da cardinalidade máxima do conjunto independente.*

PROBLEMA	NODOS	ARESTAS
MIS ₁	28	210
MIS ₂	70	1855
MIS ₃	120	5460
MIS ₄	496	107880
MIS ₅	45	918
MIS ₆	378	70551
MIS ₇	64	1824
MIS ₈	64	704
MIS ₉	256	31616
MIS ₁₀	256	20864
MIS ₁₁	200	14834
MIS ₁₂	171	9435
MIS ₁₃	200	1534
MIS ₁₄	200	3235
MIS ₁₅	200	8473
MIS ₁₆	300	21928
MIS ₁₇	200	13868
MIS ₁₈	200	17863

Tal como foi mencionado na secção anterior, calculou-se uma solução inicial para o problema da determinação do conjunto independente máximo, utilizando dois processos distintos. Uma primeira técnica consistiu na aplicação do método de *Lemke* à formulação (4.6). O segundo método determina um ponto estacionário do programa quadrático (4.2) utilizando o código MINOS. Na Tabela 4.2 usa-se a designação NILEMKE e VALLEMKE para indicar o número de iterações que o método de *Lemke* necessitou e a cardinalidade do conjunto independente obtido. Analogamente NIPQ e VALPQ representam as mesmas medidas de grandeza para o programa quadrático (4.2) obtidas com o código MINOS e NIND é o número de independência conhecido. Da observação da tabela, verifica-se que o método de *Lemke* é um processo eficiente para a determinação do conjunto independente máximo, tendo obtido o conjunto independente máximo em 61% dos casos testados com um número muito reduzido de iterações. Verifica-se ainda que o programa MINOS necessita sempre de um maior número

de iterações para obter o mesmo valor de função objectivo. Um resultado interessante que se constata também desta tabela, é que o número de iterações que o método de *Lemke* utilizou para determinar um conjunto independente corresponde sempre à cardinalidade do conjunto independente obtido adicionado de uma unidade. Uma conclusão semelhante foi também observada em [Massaro et al., 2002] para a formulação *MPEC* (4.5). Na experiência computacional efectuada pelos autores, foi verificado que uma variável z_i uma vez entrando na base nunca mais a abandona, o que também se verificou nas nossas experiências.

Tabela 4.2: *Determinação da primeira solução admissível para o conjunto independente máximo*

PROBLEMA	NiPQ	VALPQ	NiLEMKE	VALLEMKE	NIND
MIS ₁	6	4	5	4	4
MIS ₂	26	14	15	14	14
MIS ₃	14	8	9	8	8
MIS ₄	30	16	17	16	16
MIS ₅	16	9	10	9	16
MIS ₆	52	27	28	27	126
MIS ₇	62	32	33	32	32
MIS ₈	6	4	5	4	4
MIS ₉	253	128	129	128	128
MIS ₁₀	30	16	17	16	16
MIS ₁₁	24	13	14	13	21
MIS ₁₂	12	7	8	7	11
MIS ₁₃	22	12	13	12	12
MIS ₁₄	46	24	25	24	24
MIS ₁₅	114	58	59	58	58
MIS ₁₆	22	12	13	12	25
MIS ₁₇	24	13	14	13	18
MIS ₁₈	56	29	30	29	42

Na Tabela 4.3 apresentam-se os resultados da aplicação do algoritmo sequencial complementar na determinação do conjunto independente máximo utilizando a formulação (4.6). Nessa e nas tabelas seguintes, NIT representa o número total de iterações utilizadas pelo algoritmo, NOT o número de nodos totais percorridos e NIS e NOS representam respectivamente, o número de pivotagens e de nodos pesquisados até à obtenção da solução óptima. A sigla IT designa o número de *GLCPs*(λ_k) resolvidos pelo algoritmo sequencial e T o tempo computacional em segundos para terminar o processo. Na observação da tabela conclui-se que o algoritmo permite resolver todos os problemas à excepção de dois. Nesses dois casos, o algoritmo foi capaz de obter conjuntos independentes de cardinalidade muito próxima da máxima. É interessante notar que os autores em [Massaro et al., 2002] obtêm exactamente os mesmos conjuntos independentes. Podemos assim afirmar que este algoritmo apresenta um bom desempenho na determinação da solução óptima do problema. Conforme já se tinha

Tabela 4.3: *Determinação do MIS utilizando o algoritmo sequencial complementar*

PROBLEMA	NIS	NOS	IT	NIT	NOT	T
MIS ₁	5	0	1	4812	1231	1.02
MIS ₂	15	0	1	50016	2091	13.70
MIS ₃	9	0	1	50068	3541	19.95
MIS ₄	17	0	1	50033	929	71.15
MIS ₅	259	9	5	50261	3912	10.04
MIS ₆ ⁽¹⁾	4184	37	19	54372	437	107.34
MIS ₇	33	0	1	50013	994	10.46
MIS ₈	5	0	1	17828	3823	6.29
MIS ₉	129	0	1	50132	603	79.18
MIS ₁₀	17	0	1	50048	1030	44.41
MIS ₁₁ ⁽¹⁾	34314	891	9	84317	1960	52.46
MIS ₁₂	2312	39	6	52315	2600	18.32
MIS ₁₃	13	0	1	50031	4426	102.31
MIS ₁₄	25	0	1	50055	3140	108.15
MIS ₁₅	59	0	1	50273	222	73.21
MIS ₁₆	1847	37	17	51869	933	88.96
MIS ₁₇	11530	218	6	61537	1440	46.92
MIS ₁₈	7058	145	13	57060	1359	36.04

⁽¹⁾ O algoritmo não determinou a solução óptima neste exemplo, utilizando o número máximo de iterações permitidas no método enumerativo (50000). Mesmo aumentado esse número para 100000, não é obtida a solução óptima. As cardinalidades dos conjuntos independentes obtidas foram 125 e 20, respectivamente.

referido anteriormente, o grande esforço computacional deste método está em verificar a optimalidade da solução, que nestes exemplos não foi conseguido por se ter atingido o limite máximo imposto de pivotagens no método enumerativo.

Tal como referimos anteriormente, o método de *Lemke* é um processo muito dependente da regra de *Bland* para este tipo de problemas. Com o intuito de verificar essa relação, foi efectuada uma experiência que consistiu em aplicar o método de *Lemke* às permutações principais da matriz $(A + I)$. Designamos por permutação de ordem p a seguinte ordenação dos vértices do grafo: $(p, p + 1, \dots, n, 1, 2, \dots, p - 1)$. Apresentam-se na Tabela 4.4, os resultados das n permutações dos problemas teste em que o método de *Lemke* não determinou o conjunto independente máximo utilizando a ordem natural dos vértices. Dado que em alguns casos existe uma grande diversidade de conjuntos independentes obtidos, designamos por VALLEMKE as cardinalidades dos três melhores conjuntos independentes que o método de *Lemke* obteve e FREQUÊNCIA a frequência com que esses conjuntos independentes ocorreram. Para cada um destes exemplos foi seleccionada a permutação em que se obteve o maior conjunto independente e utilizado o algoritmo sequencial complementar para resolver o problema da determinação do conjunto independente máximo com a formulação (4.6) e a solução inicial do GLCP dada pelo método de *Lemke*. Os resultados desta experiência

Tabela 4.4: *Determinação de conjuntos independentes pelo método de Lemke utilizando n permutações principais*

PROBLEMA	VAL LEMKE	FREQUÊNCIA
MIS ₅	16	22
	15	6
	14	10
MIS ₆	125	70
	124	2
	123	4
MIS ₁₁	18	1
	17	7
	16	18
MIS ₁₂	9	42
	8	21
	7	108
MIS ₁₆	22	1
	20	2
	18	6
MIS ₁₇	16	1
	15	3
	14	23
MIS ₁₈	35	4
	34	2
	33	7

apresentam-se na Tabela 4.5. Salienta-se que o algoritmo foi capaz de obter um conjunto

Tabela 4.5: *Determinação do conjunto independente máximo pelo algoritmo sequencial complementar*

PROBLEMA	NIS	NOS	IT	NIT	NOT	T
MIS ₅	17	0	1	50035	5687	19.85
MIS ₆	33249	546	2	83252	1326	173.26
MIS ₁₁	3518	872	5	85129	1939	66.70
MIS ₁₂	221	7	4	50300	2035	45.29
MIS ₁₆	261	7	4	50315	745	80.90
MIS ₁₇	558	12	4	50583	1269	40.22
MIS ₁₈	9019	204	9	59019	1238	38.64

independente máximo em todos os casos, apesar de não ter sido possível mostrar que as soluções ótimas foram obtidas ($NIT > 50000$). Esses resultados indicam que será vantajoso desenvolver um processo que inicialmente utilize o método de *Lemke* para algumas permutações principais da matriz A , para depois usar o melhor conjunto independente obtido nessa fase como solução inicial para o método sequencial complementar.

A formulação *MPEC* associada ao problema de dois níveis foi também investigada, tendo o problema (4.9) sido resolvido pelo algoritmo sequencial complementar. Tal como já foi afirmado anteriormente, esta formulação conduz a problemas de dimensões elevadas, pelo que não foi possível resolver grande parte dos exemplos de teste. Apresentam-se na Tabela 4.6 os resultados obtidos para os problemas em que foi possível aplicar o algoritmo. As conclusões afirmadas na formulação anterior a propósito do esforço computacional do método para a verificação da optimalidade, intensificam-se com esta formulação. De facto, para a maioria dos problemas teste o algoritmo não conseguiu terminar por ter gerado árvores de grandes dimensões no método enumerativo. Conclui-se assim que esta formulação é claramente pior do que a anterior para a determinação do conjunto independente máximo, pelo que não é de usar na prática.

Tabela 4.6: *Algoritmo sequencial complementar para a resolução do MPEC (4.9).*

PROBLEMA	NIS	NOS	NIT	NOT	T
MIS ₁	5	0	23288	7870	14.56
MIS ₅	187	25	<i>a)</i>		10.01
MIS ₆ ⁽¹⁾	1282	133	<i>a)</i>		112.23
MIS ₁₁ ⁽¹⁾	4841	83	<i>a)</i>		146.34
MIS ₁₂	42556	3528	<i>a)</i>		145.04
MIS ₁₇ ⁽¹⁾	77254	1438	<i>a)</i>		229.23

⁽¹⁾ O algoritmo não conseguiu determinar a solução óptima neste exemplo. As cardinalidades dos conjuntos independentes obtidas foram 12, 17 e 16 respectivamente.

a) O algoritmo apresenta uma lista de nós abertos de dimensão muito elevada (superior a 10000), pelo que é interrompido.

Finalmente testou-se a formulação (4.5) do problema *MIS* para determinar o conjunto independente máximo dos problemas teste. Em três dos problemas foi possível terminar o processo, embora sem conseguir provar que a solução obtida era de facto a óptima. Nos restantes casos, o algoritmo foi interrompido por terem sido utilizadas mais de 10000 iterações na determinação de uma solução admissível para o *GLCP*. No entanto, resultados computacionais apresentados em [Massaro et al., 2002] indicam que esta formulação quadrática associada ao *MPEC* (4.5) é interessante para o cálculo do número de independência. Podemos por isso concluir que, para este tipo de formulação *MPEC*, o algoritmo sequencial complementar tem um mau desempenho, o que deve levar no futuro à procura de outras formas de resolver esse *MPEC*. Como referimos anteriormente, esse problema tem uma forma paramétrica com um só parâmetro, pelo que um processo de optimização paramétrica complementar poderá ser útil neste caso.

Como conclusão final deste estudo computacional, podemos afirmar que à semelhança das

conclusões apresentadas em [Massaro et al., 2002] para a formulação quadrática (4.2), o método de *Lemke* parece ser bastante útil na determinação de um conjunto independente para um grafo a partir da formulação *MPEC* (4.6). A utilização do método sequencial complementar para a resolução desse *MPEC* parece também ser interessante. A estrutura desse *MPEC* deve no futuro ser devidamente explorada de modo a tornar esse processo sequencial complementar ainda mais eficaz para a determinação do conjunto independente máximo a partir da sua formulação como um *MPEC*.

Capítulo 5

Estimação do número de condição

Neste capítulo é estudado o problema da estimação do número de condição de uma matriz a partir da utilização de técnicas de programação matemática com restrições de equilíbrio.

5.1 Definição do Problema

O cálculo do número de condição de uma matriz é um dos temas clássicos de análise numérica. Para problemas em que os resultados variam continuamente em função dos dados, esse valor mede a sensibilidade do problema a pequenas perturbações nos dados. O número de condição é entendido simultaneamente como um parâmetro limite para a complexidade e para a análise de erros de aproximação na resolução de sistemas. A imprecisão nos dados está sempre presente em processos que exigem medições e, dado que os computadores têm precisão finita, é importante conhecer o impacto que as propriedades da matriz em causa podem induzir nos resultados, independentemente dos erros de arredondamento associados ao algoritmo de resolução ou ao computador utilizado.

A precisão numérica da solução de um sistema de equações lineares e a convergência do método de resolução estão relacionadas com a singularidade ou quase singularidade da matriz que o define. Para quantificar a noção de quase singularidade utiliza-se o número de condição, sendo bem condicionada uma matriz com um número de condição próximo de um e mal condicionada ou quase singular se esse número é muito superior a um. Deste modo, facilmente se conclui a importância de conhecer o número de condição de uma matriz. No entanto para matrizes de dimensão elevada, o cálculo do seu valor exacto é impraticável, pelo facto do número de operações envolvidas ser superior ao número de operações necessárias para a resolução do sistema. Por isso têm sido utilizados na literatura vários processos para a estimação do número de condição, dos quais se destacam os estimadores de *Hager* [Hager, 1984] e o estimador de *Higham* [Higham, 1988]. Para introduzir estes estimadores

apresentam-se os seguintes conceitos.

Definição 5.1 A norma l_1 de uma matriz quadrada $A \in \mathbb{R}^{n \times n}$ é dada por

$$\|A\|_1 = \max_{x \in \mathbb{R}^n \setminus \{0\}} \frac{\|Ax\|_1}{\|x\|_1} \quad (5.1)$$

$$\text{com } \|x\|_1 = \sum_{j=1}^n |x_j|.$$

Da definição anterior resulta que $\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$.

Definição 5.2 O número de condição de uma matriz A invertível para a norma l_1 é o valor

$$\text{cond}_1(A) = \|A\|_1 \|A^{-1}\|_1$$

com A^{-1} a inversa de A .

Os estimadores para o cálculo do número de condição de uma matriz procuram determinar um valor aproximado de $\|A^{-1}\|_1$ sem determinar a matriz inversa de A . A descrição dos algoritmos para a determinação da norma da matriz será realizada para a $\|A\|_1$ podendo ser utilizados para estimar a norma l_∞ quando aplicados à matriz transposta, dado que a norma l_∞ de uma matriz $A \in \mathbb{R}^{n \times n}$ satisfaz

$$\|A\|_\infty = \|A^T\|_1.$$

5.2 Estimadores de *Hager* e de *Higham*

Considerando uma matriz $A \in \mathbb{R}^{n \times n}$, a norma $\|A\|_1$ pode ser determinada a partir do programa

$$\begin{aligned} \text{Maximizar } F(x) = \|Ax\|_1 &= \sum_{i=1}^n \left| \sum_{j=1}^n a_{ij} x_j \right| \\ \text{sujeito a } & x \in K \end{aligned} \quad (5.2)$$

com

$$K = \{x \in \mathbb{R}^n : \|x\|_1 \leq 1\}.$$

Antes de apresentarmos o processo de resolver este programa, devemos distinguir as seguintes propriedades [Higham, 1988].

- (i) F é diferenciável para todos os pontos $x \in K$ tais que $(Ax)_i \neq 0, \forall i$.

(ii) Se $y \in K$ tal que $\|y - x\|$ é suficientemente pequeno, então tem-se

$$F(y) = F(x) + \nabla F(x)^T(y - x), \quad (5.3)$$

com $\nabla F(x)$ o vector gradiente de F e $x \in K$ tal que $(Ax)_i \neq 0, \forall i$. Além disso

$$|\nabla F(x)^T y| \leq \|\nabla F(x)\|_\infty \|y\|_1 \leq \|\nabla F(x)\|_\infty,$$

portanto a verificação da condição

$$\|\nabla F(x)\|_\infty \leq \nabla F(x)^T x \quad (5.4)$$

estabelece que x é um máximo local de F em K .

(iii) Como a função F definida por (5.2) é convexa, então o máximo global de F é obtido num dos vértices e^j do conjunto convexo K .

O algoritmo de *Hager* é um algoritmo local que determina um máximo local para o problema de optimização (5.2). O algoritmo procura investigar apenas vectores da base canónica, com excepção do ponto inicial e move-se através desses vectores usando direcções de pesquisa baseadas no gradiente de F referido anteriormente. A desigualdade (5.4) fornece um critério de paragem para o algoritmo. Os passos do algoritmo são apresentados a seguir, onde usamos a notação $\text{sign}(y)$ para representar o vector com componentes definidas por

$$(\text{sign}(y))_i = \begin{cases} 1 & \text{se } y_i \geq 0 \\ -1 & \text{se } y_i < 0. \end{cases}$$

ALGORITMO DE HAGER PARA ESTIMAR $\|A\|_1$

Passo 0: *Inicialização*

Seja $x = (\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$.

Passo 1: *Calculo do gradiente* $\nabla F(z)$

Calcule $y = Ax$,
 $\xi = \text{sign}(y)$,
 $z = A^T \xi$.

Passo 2: *Critério de paragem*

Se $\|z\|_\infty \leq z^T x$, termine com $\gamma = \|y\|_1$.

Passo 3: *Direcção de pesquisa*

Seja r tal que $\|z\|_\infty = |z_r| = \max_{1 \leq i \leq n} |z_i|$. Faça $x = e^r$ e volte ao Passo 1.

Este processo é particularmente útil para a estimação do número de condição de uma matriz, uma vez que o algoritmo pode ser usado para calcular $\|B^{-1}\|_1$. De facto sendo $A = B^{-1}$,

então no Passo 1 tem-se

$$\begin{aligned} y &= B^{-1}x &\Leftrightarrow & By = x \\ z &= (B^{-1})^T \xi &\Leftrightarrow & B^T z = \xi. \end{aligned}$$

Portanto a norma de B^{-1} pode ser determinada pelo algoritmo de *Hager*. Em cada iteração do processo é necessária a resolução de dois sistemas com as matrizes B e B^T . Se a decomposição LU de B for conhecida, então cada iteração corresponde à resolução de quatro sistemas triangulares.

O algoritmo de *Higham* utiliza passos semelhantes aos do algoritmo de *Hager* um número finito de vezes. Tal como é descrito em [Higham, 1988], a vasta experiência computacional efectuada pelo autor permitiu concluir que o número de iterações usadas pelo algoritmo até encontrar um ponto estacionário é bastante pequeno, pelo que o autor impõe um número máximo de iterações igual a cinco. Além disso o teste de convergência do algoritmo deve ser efectuado apenas em iterações superiores à primeira. Outros melhoramentos ao nível da convergência do algoritmo consistem em parar o algoritmo se houver repetição do vector ξ em iterações sucessivas ou se não se verificar um incremento no valor da estimativa.

Os passos do algoritmo para o caso particular de um ponto inicial resumem-se a seguir.

ALGORITMO DE HIGHAM PARA ESTIMAR $\|A\|_1$

Passo 0: *Inicialização*

Seja x um ponto do conjunto K com $\|x\|_1 = 1$.

Passo 1: *Cálculo do gradiente inicial*

$$\begin{aligned} \text{Calcule} \quad y &= Ax \\ \gamma &= \|y\|_1 \\ \xi &= \text{sign}(y) \\ z &= A^T \xi. \end{aligned}$$

Faça $k = 2$.

Passo 2: *Critério de paragem*

$$\begin{aligned} \text{Repita:} \quad r &= \min\{i : |z_i| = \|z\|_\infty\} \\ y &= Ae^r \\ \bar{\gamma} &= \gamma \\ \gamma &= \|y\|_1 \\ \text{Se } \text{sign}(y) &= \xi \text{ ou } \gamma \leq \bar{\gamma} \text{ pare com } \bar{\gamma} \text{ a estimativa pretendida. De outro modo} \\ \xi &= \text{sign}(y) \\ z &= A^T \xi \\ k &= k + 1 \\ \text{até} \quad \|z\|_\infty &= z_r \text{ ou } k > 5. \end{aligned}$$

Neste processo, o ponto inicial é um vector cujas componentes são geradas aleatoriamente com distribuição uniforme $[0, 1]$. Apesar de bastante robusto, este processo apresenta ainda problemas na estimação do número de condição para certo tipo de matrizes [Higham, 1988]. Tal como anteriormente, o algoritmo permite determinar uma estimativa de $\|B^{-1}\|_1$ resolvendo em cada iteração dois sistemas lineares com B e B^T .

A utilização de vários pontos iniciais permite aumentar a eficiência do algoritmo, conforme é referido em [Higham e Tisseur, 2000]. A eficiência do processo nesta versão de pontos iniciais em bloco é aumentada pelo facto de se manter a informação das colunas da matriz ξ anteriormente usadas e consequente alteração aleatória dessas colunas quando os cálculos forem redundantes. A implementação computacional deste processo, considera por defeito dois pontos iniciais, permitindo no entanto escolher até n pontos iniciais, naturalmente com um aumento do esforço computacional.

5.3 Formulação *MPEC*

Para obter a formulação *MPEC* do problema, substituímos o conjunto K referido na secção anterior pelo simplex

$$\tilde{K} = \{x \in \mathbb{R}^n : e^T x = 1, x \geq 0\}$$

com $e = [1 \ 1 \ \dots \ 1]^T$. Então o programa não linear para o cálculo da $\|A\|_1$ pode ser definido por

$$\begin{aligned} \text{Maximizar} \quad & F(x) = \|Ax\|_1 = \sum_{i=1}^n \left| \sum_{j=1}^n a_{ij} x_j \right| \\ \text{sujeito a} \quad & e^T x = 1 \\ & x \geq 0. \end{aligned} \tag{5.5}$$

Este problema de optimização é equivalente ao programa (5.2). De facto \tilde{K} tem como pontos extremos (e^1, e^2, \dots, e^n) e o máximo da função F é alcançado num desses pontos, ou seja,

$$\exists r \in \{1, \dots, n\} : F(e^r) = \max_{x \in \tilde{K}} F(x).$$

Por outro lado,

$$\exists r \in \{1, \dots, n\} : \|A\|_1 = \|Ae^r\|_1.$$

Como todos os vectores da base canónica pertencem aos dois conjuntos convexos K e \tilde{K} , conclui-se que os programas (5.2) e (5.5) são equivalentes. Tal como anteriormente, a dificuldade de resolução deste problema prende-se com o facto de (5.5) ser um programa côncavo. No entanto se a matriz tiver elementos não negativos é possível resolver facilmente esse programa, como se mostra a seguir. Com efeito, como $A \geq 0$ e $x \geq 0$, então

$$F(x) = \sum_{i=1}^n \left| \sum_{j=1}^n a_{ij} x_j \right| = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_j = \sum_{j=1}^n \left(\sum_{i=1}^n a_{ij} \right) x_j$$

ou seja,

$$F(x) = d^T x$$

com

$$d = A^T e.$$

Assim o programa (5.5) reduz-se ao seguinte programa linear

$$\begin{array}{ll} \text{Maximizar} & d^T x \\ \text{sujeito a} & e^T x = 1 \\ & x \geq 0 \end{array}$$

que tem como solução óptima

$$x = e^r, \quad \text{com} \quad d_r = \max\{d_i : i = 1, \dots, n\}.$$

Portanto

$$\|A\|_1 = d_r.$$

No caso geral, considere-se a seguinte substituição

$$\sum_{j=1}^n a_{ij} x_j = w_i - z_i$$

com

$$\left. \begin{array}{l} w_i \geq 0 \\ z_i \geq 0 \\ w_i z_i = 0 \end{array} \right\} \forall i = 1, \dots, n.$$

Deste modo

$$\left| \sum_{j=1}^n a_{ij} x_j \right| = w_i + z_i$$

e o programa (5.5) é equivalente ao problema

$$\begin{array}{ll} \text{Maximizar} & \sum_{i=1}^n w_i + z_i \\ \text{sujeito a} & \sum_{j=1}^n a_{ij} x_j = w_i - z_i, \quad i = 1, \dots, n \\ & e^T x = 1 \\ & x \geq 0, z \geq 0, w \geq 0 \\ & z^T w = 0. \end{array}$$

Portanto a determinação da norma l_1 de A reduz-se ao seguinte *MPEC*

$$\begin{aligned}
 & \text{Maximizar} && e^T w + e^T z \\
 & \text{sujeito a} && Ax = w - z \\
 & && e^T x = 1 \\
 & && x \geq 0, z \geq 0, w \geq 0 \\
 & && z^T w = 0.
 \end{aligned} \tag{5.6}$$

Tal como nos algoritmos anteriores, a importância deste processo advém do facto de se poder aplicar na estimação do número de condição de uma matriz. Para esse objectivo há que determinar um valor aproximado de $\|A^{-1}\|_1$, devendo ser considerados dois casos.

Caso Particular: A é uma matriz Minkowski, isto é, $A \in P$ e os elementos não diagonais são todos não negativos. Então $A^{-1} \geq 0$ [Cottle et al., 1992] e tem-se

$$\|A^{-1}\|_1 = d_r = \max\{d_i, i = 1, \dots, n\},$$

com d o vector solução do sistema

$$A^T d = e. \tag{5.7}$$

É de notar que este tipo de matrizes aparece muito frequentemente na resolução de problemas de equações de derivadas parciais elípticas pelo método dos elementos finitos e diferenças finitas [Johnson, 1990]. Chegamos assim à interessante conclusão que o número de condição de uma matriz Minkowski se pode estimar a partir da resolução do sistema (5.7) e tem-se

$$\text{cond}_1(A) \simeq \|A\|_1 d_r$$

com d o vector solução desse sistema.

Caso Geral: De acordo com o *MPEC* (5.6), podemos determinar uma estimativa de $\|A^{-1}\|_1$ a partir da resolução do programa

$$\begin{aligned}
 & \text{Minimizar} && -e^T w - e^T z \\
 & \text{sujeito a} && A^{-1}x = w - z \\
 & && e^T x = 1 \\
 & && x \geq 0, z \geq 0, w \geq 0 \\
 & && z^T w = 0.
 \end{aligned} \tag{5.8}$$

Para determinar uma solução do correspondente *GLCP*, seja \bar{x} o vector da base canónica, tal que

$$\bar{x} = e^r = [0 \dots 0 \underbrace{1}_r 0 \dots 0]^T$$

para certo valor de $r \in \{1, \dots, n\}$. Resolvendo o sistema

$$Au = \bar{x}$$

determina-se

$$u = A^{-1}\bar{x}.$$

Para completar o cálculo da solução do *GLCP*, basta usar as relações

$$\begin{cases} \bar{w}_i = \bar{u}_i \text{ e } \bar{z}_i = 0, & \text{se } u_i \geq 0 \\ \bar{w}_i = 0 \text{ e } \bar{z}_i = -\bar{u}_i, & \text{se } u_i < 0. \end{cases}$$

Uma vez obtida a solução do *GLCP*, é usado um algoritmo para a determinação de um ponto estacionário do *MPEC*. Esse processo é iniciado com a solução do *GLCP* obtida e efectua trocas entre os vectores $\bar{x} = e^r$ e $\tilde{x} = e^p$, garantindo deste modo a complementaridade da solução. Descrevem-se seguidamente os cálculos inerentes a uma operação deste tipo.

Seja B a base correspondente à solução do primeiro *GLCP* definida por

$$B = \begin{bmatrix} \pm I & A^{-1}e^r \\ 0 & 1 \end{bmatrix},$$

com $\pm I$ correspondendo a uma coluna da matriz identidade de ordem n ou à sua coluna simétrica, consoante a variável básica associada for z_i ou w_i , respectivamente. Considerando $\begin{bmatrix} \pi & \gamma \end{bmatrix}^T \in \mathbb{R}^{n+1}$ o vector das variáveis duais deste problema, a solução dual é obtida pela resolução do sistema

$$\begin{bmatrix} \pm I & 0 \\ (A^{-1}e^r)^T & 1 \end{bmatrix} \begin{bmatrix} \pi \\ \gamma \end{bmatrix} = \begin{bmatrix} -e \\ 0 \end{bmatrix}.$$

Portanto

$$\begin{cases} \pi_i = 1, & \text{se } u_i \geq 0 \\ \pi_i = -1, & \text{se } u_i < 0 \end{cases}$$

e γ calculado pela equação

$$\gamma = -u^T \pi.$$

Assim sendo, os coeficientes de custo reduzido associados às variáveis não básicas x_j , $j \in \{1, \dots, n\} \setminus \{r\}$ podem ser obtidos pela seguinte equação

$$\bar{c}_j = 0 - \begin{bmatrix} \pi \\ \gamma \end{bmatrix}^T \begin{bmatrix} A_{\cdot j}^{-1} \\ 1 \end{bmatrix} = -\gamma - \pi^T A_{\cdot j}^{-1}.$$

Se v é o vector definido por

$$v = \pi^T A^{-1},$$

o custo reduzido \bar{c}_j associado a cada variável não básica x_j pode ser reescrito da forma

$$\bar{c}_j = -\gamma - v_j,$$

com v_j determinado pela resolução do sistema

$$A^T v = \pi.$$

Se L é o conjunto dos índices das variáveis não básicas associados às variáveis x_j com $j \neq r$, a optimalidade do vector $[\pi \ \gamma]^T$ pode facilmente ser verificada. Com efeito, se $\bar{c}_j \geq 0 \forall j \in L$, o algoritmo termina com um ponto estacionário do programa (5.8). Caso contrário deve ser efectuada uma nova troca.

Os passos do algoritmo são apresentados a seguir.

ESTIMADOR MPEC PARA O CÁLCULO DE $\|A^{-1}\|_1$

Passo 0: *Inicialização*

Seja $x = e^r$ um vector arbitrário da base canónica.

Passo 1: *Determinação de um ponto estacionário*

Calcule $Au = x$

Se $u_i \geq 0 \Rightarrow \pi_i = 1$

Se $u_i < 0 \Rightarrow \pi_i = -1$

$\gamma = -u^T \pi$

$A^T v = \pi$

$\bar{c} = -\gamma - v.$

Passo 2: *Critério de paragem*

Se $\bar{c}_j \geq 0 \forall j \in L$, pare o algoritmo com $est = -\gamma$ a estimativa pretendida.

Caso contrário, seja $\bar{c}_s = \min\{\bar{c}_j : \bar{c}_j < 0 \text{ e } j \in L\}$. Faça $x = e^s$ e volte ao Passo 1.

A experiência computacional efectuada permitiu concluir que o uso do vector e^1 da base canónica na inicialização deste processo conduz a bons estimadores com um máximo de três iterações para todos os tipos de matrizes testadas. Uma melhoria na eficiência do processo consiste em utilizar vectores iniciais em bloco, por analogia ao algoritmo de melhoramento de *Higham*. A informação do histórico de vectores da base canónica utilizados, conjuntamente com a avaliação da melhoria da estimativa relativamente à iteração anterior, são critérios bastante eficientes para a obtenção de boas estimativas precocemente. De facto, utilizando inicialmente t vectores da base canónica, no final de cada iteração são sugeridos $p \leq t$ novos vectores de análise, que serão sujeitos a um teste de não repetição de vectores entre os

Tipo IV. Matrizes Pentadiagonais com elemento genérico m_{ij} tal que

$$\begin{aligned} m_{ii} &= 6 \\ m_{i,i-1} &= m_{i-1,i} = -4 \\ m_{i,i-2} &= m_{i-2,i} = 1 \end{aligned}$$

e todos os restantes elementos são iguais a zero.

Tipo V. Matrizes de *Murty* [Murty, 1988] que apresentam a seguinte estrutura

$$\begin{bmatrix} 1 & & & & \\ 2 & 1 & & & \\ 2 & 2 & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ 2 & 2 & 2 & \dots & 1 \end{bmatrix}.$$

Tipo VI. Matrizes de *Fathy* [Murty, 1988] da forma

$$F = M^T M$$

com M uma matriz de *Murty*.

Tipo VII. Matrizes cuja inversa se pode escrever

$$A^{-1} = I + \theta * C$$

com C uma matriz arbitrária tal que

$$Ce = C^T e = 0,$$

para $e = [1 \ 1 \ \dots \ 1]^T$ e o parâmetro θ a tomar valores próximos de zero.

Tipo VIII. Matrizes da colecção **MatrixMarket** [Boisvert et al., 1998] que engloba o conjunto de matrizes da **Harwell-Boeing**. Na Tabela 5.1 apresentam-se algumas especificações das matrizes seleccionadas.

A Tabela 5.2 apresenta os resultados da aplicação dos vários algoritmos a este tipo de matrizes, utilizando um vector inicial para o caso do algoritmo de *Higham* e do método que explora a formulação *MPEC*.

Foram efectuadas diversas experiências de carácter heurístico com o novo algoritmo que permitiram escolher o vector e^r da base canónica para inicializar o processo. De entre as várias possibilidades concluiu-se que o vector e^1 conduzia na maioria das matrizes testadas a melhores estimativas do valor de $\|A^{-1}\|_1$, pelo que foi o vector utilizado nos resultados computacionais que se seguem.

Tabela 5.1: *Propriedades das matrizes Tipo VIII.*

			Nº ELEMENTOS	
NOME	ORIGEM	ORDEM	NÃO NULOS	TIPO
MATRIX ₁	Elementos Finitos	5489	143300	SPD
MATRIX ₂	Elementos Finitos	5489	143300	SPD
MATRIX ₃	Elementos Finitos	5489	143300	SPD
MATRIX ₄	Elementos Finitos	5489	112505	SPD
MATRIX ₅	Elementos Finitos	5489	112505	SPD
MATRIX ₆	Elementos Finitos	5489	112505	SPD
MATRIX ₇	Engenharia Estrutural	1806	32630	SPD
MATRIX ₈	Engenharia Estrutural	3948	60882	SPD
MATRIX ₉	Engenharia Estrutural	1224	28675	SPD
MATRIX ₁₀	Engenharia Estrutural	4410	111717	Indefinida
MATRIX ₁₁	Modelação Económica	2529	90158	Não simétrica
MATRIX ₁₂	Modelação Petrolífera	3312	20793	Não simétrica
MATRIX ₁₃	Diferenças Finitas	729	2673	SPD
MATRIX ₁₄	Diferenças Finitas	957	2547	SPD
MATRIX ₁₅	Dinâmica de fluídos	4241	131556	Não simétrica
MATRIX ₁₆	Modelação de elementos Finitos	9152	765944	Não simétrica
MATRIX ₁₇	Modelação de elementos Finitos	3565	67591	Não simétrica

Tabela 5.2: *Determinação da $\|A^{-1}\|_1$ utilizando os vários algoritmos, com um só vector inicial.*

MATRIZ		ESTIMADOR DE HAGER			ESTIMADOR DE HIGHAM			ESTIMADOR MPEC			NÚMERO DE CONDIÇÃO
		$\ A^{-1}\ _1$	NIT	T	$\ A^{-1}\ _1$	NIT	T	$\ A^{-1}\ _1$	NIT	T	
TIPO I	N										
	50	2.89E+00	2	0.00	2.89E+00	2	0.01	2.89E+00	1	0.00	2.31E+01
	250	3.00E+00	2	0.01	3.00E+00	2	0.01	3.00E+00	1	0.00	2.40E+01
	500	3.00E+00	2	0.01	3.00E+00	2	0.02	3.00E+00	1	0.00	2.40E+01
	1000	3.00E+00	2	0.03	3.00E+00	2	0.03	3.00E+00	1	0.00	2.40E+01
	2000	3.00E+00	2	0.06	3.00E+00	2	0.05	3.00E+00	1	0.01	2.40E+01
	4000	3.00E+00	2	0.13	3.00E+00	2	0.09	3.00E+00	1	0.02	2.40E+01
TIPO II	α (N=4000)										
	0.50	4.00E+00	2	48.52	4.00E+00	2	37.40	4.00E+00	1	54.35	1.60E+04
	0.25	8.00E+00	2	46.63	8.00E+00	2	39.21	8.00E+00	1	62.14	3.20E+04
	0.125	1.60E+01	2	37.19	1.60E+01	2	66.92	1.60E+01	1	59.94	6.40E+04
	0.01	2.00E+02	2	37.06	2.00E+02	2	58.33	2.00E+02	1	74.87	8.00E+05
	1.0E-03	2.00E+03	2	46.00	2.00E+03	2	37.88	2.00E+03	1	84.88	8.00E+06
	1.0E-04	2.00E+04	2	44.22	2.00E+04	2	55.43	2.00E+04	1	54.34	8.00E+07
	1.0E-05	2.00E+05	2	61.10	2.00E+05	2	49.68	2.00E+05	1	61.89	8.00E+08

Tabela 5.2: (continuação)

MATRIZ		ESTIMADOR DE HAGER			ESTIMADOR DE HIGHAM			ESTIMADOR MPEC			NÚMERO DE CONDIÇÃO
		$\ A^{-1}\ _1$	NIT	T	$\ A^{-1}\ _1$	NIT	T	$\ A^{-1}\ _1$	NIT	T	
TIPO III	N										
	50	4.90E+01	2	0.02	1.00E+00	2	0.13	5.00E+01	1	0.02	1.00E+02
	250	2.49E+02	2	0.00	1.00E+00	2	0.01	2.50E+02	1	0.01	5.00E+02
	500	4.99E+02	2	0.01	1.00E+00	2	0.01	5.00E+02	1	0.01	1.00E+03
	1000	9.99E+02	2	0.02	1.00E+00	2	0.02	1.00E+03	1	0.03	2.00E+03
	2000	2.00E+03	2	0.03	1.00E+00	2	0.02	2.00E+03	1	0.06	4.00E+03
	4000	4.00E+03	2	0.13	1.00E+00	2	0.07	4.00E+03	1	0.24	8.00E+03
TIPO IV	N										
	50	1.90E+04	2	0.01	1.90E+04	2	0.30	1.90E+04	2	0.02	3.04E+05
	250	1.05E+07	2	0.00	1.05E+07	2	0.02	1.05E+07	2	0.01	1.68E+08
	500	1.65E+08	2	0.01	1.65E+08	2	0.01	1.65E+08	2	0.01	2.65E+09
	1000	2.63E+09	2	0.02	2.63E+09	2	0.02	2.63E+09	2	0.03	4.20E+10
	2000	4.18E+10	2	0.03	4.18E+10	2	0.04	4.18E+10	2	0.06	6.69E+11
	4000	6.68E+11	2	0.07	6.68E+11	2	0.07	6.68E+11	2	0.14	1.07E+13
TIPO V	N										
	50	9.90E+01	2	0.00	9.90E+01	2	0.01	9.90E+01	1	0.15	9.80E+03
	250	4.99E+02	2	0.04	4.99E+02	2	0.05	4.99E+02	1	0.01	2.49E+05
	500	9.99E+02	2	0.22	9.99E+02	2	0.22	9.99E+02	1	0.01	9.98E+05
	1000	2.00E+03	2	1.31	2.00E+03	2	1.32	2.00E+03	1	0.03	4.00E+06
	2000	4.00E+03	2	8.87	4.00E+03	2	9.23	4.00E+03	1	0.06	1.60E+07
	4000	8.00E+03	2	51.96	8.00E+03	2	43.63	8.00E+03	1	0.17	6.40E+07
TIPO VI	N										
	50	5.00E+03	2	0.00	5.00E+03	2	0.01	5.00E+03	2	0.04	2.50E+07
	250	1.25E+05	2	0.05	1.25E+05	2	0.05	1.25E+05	2	0.01	1.56E+10
	500	5.00E+05	2	0.24	5.00E+05	2	0.24	5.00E+05	2	0.02	2.50E+11
	1000	2.00E+06	2	1.00	2.00E+06	2	1.40	2.00E+06	2	0.03	4.00E+12
	2000	8.00E+06	2	6.53	8.00E+06	2	5.32	8.00E+06	2	0.04	6.40E+13
	4000	3.20E+07	2	42.68	3.20E+07	2	67.72	3.20E+07	2	0.17	1.02E+15
TIPO VII	θ	(N=4000)									
	0.50	1.54E+03	3	38.80	1.54E+03	3	47.05	2.30E+03	2	124.15	8.14E+13
	0.25	1.34E+03	3	38.78	1.34E+03	3	73.23	1.55E+03	2	172.58	2.75E+13
	0.125	1.14E+03	4	49.57	1.14E+03	4	73.41	1.24E+03	2	153.73	1.10E+13
	0.01	2.36E+02	3	38.61	2.36E+02	3	39.98	3.02E+02	1	118.56	2.14E+11
	1.0E-03	8.02E+01	3	38.76	8.02E+01	3	42.34	9.87E+01	1	132.30	6.99E+09
	1.0E-04	1.80E+01	3	50.04	1.80E+01	3	40.01	4.09E+01	1	71.13	2.90E+08
	1.0E-05	1.01E+01	4	71.03	1.01E+01	4	43.80	1.14E+01	1	135.69	8.15E+06

Tabela 5.2: (continuação)

MATRIZ	NOME	ESTIMADOR DE HAGER			ESTIMADOR DE HIGHAM			ESTIMADOR MPEC			NÚMERO DE CONDIÇÃO
		$\ A^{-1}\ _1$	NIT	T	$\ A^{-1}\ _1$	NIT	T	$\ A^{-1}\ _1$	NIT	T	
TIPO VIII	MATRIX ₁	1.41E-01	2	4.29	1.41E-01	2	6.30	1.41E-01	2	4.55	7.47E+04
	MATRIX ₂	1.45E+01	2	2.96	1.45E+01	2	6.20	1.45E+01	2	2.49	7.66E+05
	MATRIX ₃	8.62E+03	3	5.31	8.62E+03	3	8.33	7.43E+03	2	2.54	4.54E+07
	MATRIX ₄	5.32E-02	2	18.01	5.32E-02	2	22.36	5.32E-02	2	17.46	4.93E+04
	MATRIX ₅	1.09E+01	2	2.97	1.09E+01	2	6.24	1.09E+01	2	2.57	9.22E+05
	MATRIX ₆	5.29E+03	2	2.92	5.29E+03	2	6.29	5.57E+03	2	2.45	4.62E+07
	MATRIX ₇	1.00E+00	2	0.36	1.00E+00	2	0.92	1.00E+00	2	0.32	1.11E+10
	MATRIX ₈	1.00E+00	2	2.03	1.00E+00	2	4.72	1.00E+00	2	1.84	7.66E+09
	MATRIX ₉	4.35E-03	2	0.07	4.35E-03	2	0.20	4.23E-03	2	0.08	1.23E+04
	MATRIX ₁₀	5.26E-05	2	0.15	5.26E-05	2	0.27	4.02E-05	2	0.21	4.49E+04
	MATRIX ₁₁	9.58E+03	2	1.44	9.58E+03	2	3.52	9.58E+03	2	2.72	1.00E+07
	MATRIX ₁₂	9.26E+01	2	2.46	9.26E+01	2	3.34	9.26E+01	2	2.47	3.90E+05
	MATRIX ₁₃	3.33E+01	2	0.01	3.33E+01	2	0.01	3.33E+01	2	0.02	3.00E+08
	MATRIX ₁₄	3.13E-06	2	0.01	3.12E-06	2	0.02	3.13E-06	1	0.02	3.71E+05
	MATRIX ₁₅	7.38E+07	2	3.14	7.38E+07	2	6.25	7.78E+06	2	4.01	1.84E+10
	MATRIX ₁₆	1.06E+03	2	11.05	1.06E+03	2	16.70	1.06E+03	2	15.71	3.05E+10
	MATRIX ₁₇	1.00E+00	2	0.76	1.00E+00	2	0.75	1.00E+00	1	1.03	2.26E+02

De salientar ainda que para as matrizes de *Tipo II e VII* apenas se apresentam os resultados para a maior dimensão de matriz, pois consideram-se os mais representativos da classe. Na notação usada nas tabelas, NIT representa o número de iterações que o algoritmo usou na obtenção da estimativa pretendida e T representa o tempo computacional em segundos para esse cálculo. Além disso os valores das estimativas de $\|A^{-1}\|_1$ obtidos pelos algoritmos e o número de condição $cond(A) = \|A\|_1 \|A^{-1}\|_1$ estimado pelo algoritmo *MPEC* são também apresentados.

Da Tabela 5.2 destacam-se os bons resultados obtidos pelo estimador *MPEC* para quase todas as classes de matrizes. De facto, apenas três matrizes do *Tipo VIII* apresentam piores estimativas que o algoritmo de *Higham*. De salientar ainda que nas matrizes de *Tipo III e Tipo VII*, o estimador *MPEC* é evidentemente mais eficiente que o estimador de *Higham*, quer a nível do valor da estimativa, quer no número de iterações usadas na sua obtenção. Tal como é referido pelo autor, os resultados deste estimador são dependentes do vector aleatório usado para arranque do processo, pelo que foram efectuadas várias experiências, concluindo-se no entanto que nesta versão do método, o valor da estimativa não se altera.

A eficiência dos algoritmos de *Higham* e do estimador *MPEC* foi posteriormente avaliada com a versão de vectores iniciais em bloco, tendo-se efectuado várias experiências no sentido de estabelecer o número de vectores iniciais que se deveria usar. Convém salientar que a implementação em bloco do estimador *MPEC* inclui um melhoramento correspondente à inclusão do vector e^1 da base canónica num dos vectores iniciais do processo, face ao bom

desempenho que esta heurística apresenta. A Tabela 5.3 inclui os resultados computacionais da aplicação do estimador em bloco, usando dois vectores iniciais.

Tabela 5.3: *Determinação da $\|A^{-1}\|_1$ utilizando os algoritmos de Higham e estimador MPEC em bloco com dois vectores iniciais.*

MATRIZ		ESTIMADOR DE HIGHAM			ESTIMADOR MPEC		
		$\ A^{-1}\ _1$	NIT	T	$\ A^{-1}\ _1$	NIT	T
TIPO VII	θ	(N=4000)					
	0.50	2.05E+03	3	71.78	2.30E+03	2	111.81
	0.25	1.54E+03	2	70.01	1.55E+03	2	152.53
	0.125	1.24E+03	2	47.58	1.24E+03	2	140.67
	0.01	2.95E+02	2	40.76	3.02E+02	2	141.07
	1.0E-03	8.57E+01	2	43.08	9.87E+01	2	110.45
	1.0E-04	3.79E+01	3	71.73	4.09E+01	2	176.49
	1.0E-05	1.15E+01	2	51.14	1.14E+01 *	2	180.69
TIPO VIII	NOME						
	MATRIX ₁	1.41E-01	2	6.49	1.41E-01	2	2.67
	MATRIX ₂	1.45E+01	2	6.37	1.45E+01	2	2.59
	MATRIX ₃	8.62E+03	2	6.31	8.40E+03*	2	3.29
	MATRIX ₄	5.35E-02	2	23.28	5.84E-02	2	17.80
	MATRIX ₅	1.09E+01	2	9.62	1.09E+01	2	4.28
	MATRIX ₆	5.55E+03	2	9.88	5.57E+03	2	4.37
	MATRIX ₇	1.00E+00	2	1.43	1.00E+00	2	0.53
	MATRIX ₈	1.00E+00	2	7.13	1.00E+00	2	2.75
	MATRIX ₉	4.35E-03	3	0.46	4.27E-03*	2	0.12
	MATRIX ₁₀	5.26E-05	2	0.43	4.02E-05*	2	0.33
	MATRIX ₁₁	9.58E+03	2	5.32	9.58E+03	2	4.12
	MATRIX ₁₂	9.26E+01	2	5.06	9.26E+01	2	4.04
	MATRIX ₁₃	3.33E+01	2	0.02	3.33E+01	2	0.03
	MATRIX ₁₄	3.13E-06	2	0.03	3.13E-06	2	0.04
	MATRIX ₁₅	7.38E+07	2	9.51	7.38E+07	2	6.78
	MATRIX ₁₆	1.06E+03	2	21.25	1.06E+03	2	16.33
	MATRIX ₁₇	1.00E+00	2	0.80	1.00E+00	2	1.56

* O estimador *MPEC* não conseguiu obter o valor exacto da $\|A^{-1}\|_1$.

A análise dos resultados da Tabela 5.3 mostra que o estimador *MPEC* resolve de uma forma robusta o problema da determinação da norma l_1 da matriz inversa, apresentando melhores estimativas do que o estimador de *Higham* em dezasseis casos e piores em três.

Os resultados estatísticos do desempenho dos dois algoritmos em bloco que se apresentam na Tabela 5.4 foram obtidos com base nos estimadores da norma da inversa para as matrizes do *Tipo VII* e de *Tipo VIII* para as dimensões $n = 50, 250, 500, 1000, 2000$ e 4000 . Os testes das matrizes do *Tipo VII* englobam todos os valores de θ já referidos.

Tabela 5.4: *Resultados estatísticos da aplicação dos dois métodos em bloco.*

ERRO RELATIVO (%)	DOIS VECTORES		TRÊS VECTORES	
	EST. HIGHAM	EST. MPEC	EST. HIGHAM	EST. MPEC
MÉDIA	2.4	0.7	2.2	1.2
DESVIO PADRÃO	0.06	0.02	0.06	0.04

Esses resultados permitem concluir que o uso de dois vectores iniciais é suficiente na obtenção de boas estimativas para ambos os algoritmos, salientando-se mais uma vez a eficiência do estimador *MPEC* proposto neste trabalho. É ainda de notar que os tempos de execução do estimador *MPEC* podem em alguns casos ser melhorados com uma implementação mais optimizada do processo. Será ainda importante desenvolver um processo globalizante para a estimação do número de condição que envolva apenas a resolução de sistemas de equações lineares com a matriz dada ou a sua transposta. Finalmente, salienta-se a possibilidade de estimar o número de condição de uma matriz *Minkowski* com apenas um sistema de equações lineares. A pesquisa de outras classes de matrizes com este tipo de comportamento parece ser um tópico interessante de investigação futura.

Conclusão

Neste trabalho foi desenvolvido, implementado e testado um algoritmo enumerativo paramétrico para a determinação de um mínimo global de um Problema de Programação Matemática com Restrições de Equilíbrio ou Complementaridade (*MPEC*). Tal como o método desenvolvido em [Bard e Moore, 1990], este algoritmo tem por base um processo de ramificação e limitação. Dado que o conjunto de restrições do problema *MPEC* é um problema linear complementar, o cálculo de uma solução admissível utilizou em cada nó da árvore, um método de restrições activas para a determinação de um ponto estacionário do problema que minimiza a função complementaridade, no conjunto das restrições lineares do problema. À semelhança do método sequencial complementar desenvolvido por Júdice e Faustino [Júdice e Faustino, 1992] o algoritmo enumerativo paramétrico incorpora ainda um corte, que força a função objectivo a reduzir estritamente de valor à medida que se vão gerando os sucessivos pontos estacionários. Experiência computacional realizada com problemas *MPECs* indicou que o novo método é competitivo com outras técnicas globais para o mesmo problema. Além disso o algoritmo tem a vantagem de poder ser usado para *MPECs* com restrições lineares sem estrutura especial.

O desenvolvimento de duas novas formulações para o Problema Linear Complementar e Bilinear foi outra das contribuições deste trabalho, que permitiu reforçar a utilização dos *MPECs* no domínio da optimização global. Nesse sentido foi também analisada a importância da formulação *MPEC* para resolver Problemas de Programação Inteira 0-1. A análise comparativa com os algoritmos globais anteriormente referidos permitiu concluir que o algoritmo enumerativo paramétrico resolve os problemas inteiros de uma forma razoavelmente eficaz.

A determinação do Conjunto Independente Máximo (*MIS*) de um grafo utilizando a formulação *MPEC* do problema foi também abordada neste trabalho. O uso conjunto do método de *Lemke* e do algoritmo sequencial complementar revelou-se com um desempenho bastante bom na determinação do conjunto independente máximo. Foi ainda introduzido um algoritmo de ramificação e limitação para a resolução de uma formulação quadrática do problema, baseado na decomposição de uma função quadrática numa diferença de duas funções convexas quadráticas (DC). No entanto os resultados da aplicação deste processo não foram muito promissores.

Um algoritmo para a estimação do número de condição de uma matriz foi ainda introduzido neste trabalho, que se baseou numa formulação *MPEC* do problema. Os resultados da sua aplicação a vários tipos de matrizes e dimensões mostraram que o algoritmo é eficaz e competitivo com outros estimadores existentes na literatura. Além disso foi estabelecido que o número de condição de uma matriz de Minkowski se pode estimar eficazmente com apenas um sistema de equações lineares.

Como se conclui da leitura desta tese, surgiram várias ideias que irão certamente merecer a nossa atenção futura. Nesse sentido é fundamental a incorporação de técnicas de corte para a determinação de melhores limites inferiores no algoritmo enumerativo paramétrico de forma a torná-lo mais eficaz na resolução de problemas *MPEC*. O método enumerativo paramétrico deve ser estendido para resolver *MPECs* com uma função quadrática convexa. Neste caso um algoritmo de restrições activas ou de pontos interiores terá de ser usado em cada nó do método enumerativo para a determinação de um ponto estacionário da função quadrática complementaridade num conjunto admissível constituído por restrições lineares e uma restrição quadrática convexa.

Tal como referimos no capítulo 4, uma formulação *MPEC* paramétrica para o problema da determinação do número de independência poderá conduzir a um processo de optimização paramétrica complementar para este tipo de problemas. A estrutura do *MPEC* deve também ser devidamente explorada de modo a tornar o processo sequencial complementar ainda mais eficaz para a determinação do conjunto independente máximo a partir da sua formulação como um *MPEC*.

Finalmente será também interessante investigar o desenvolvimento de técnicas globalizantes para a estimação do número de condição de uma matriz que necessitem apenas da resolução de sistemas lineares com essa matriz ou a sua transposta. O estudo de classes de matrizes que à semelhança das matrizes de Minkowski reduzem esse processo a um sistema de equações lineares, com a matriz dada ou a sua transposta, deve também ser uma área de investigação futura para a estimação eficiente do número de condição de uma matriz.

Bibliografia

- [Al-Khayyal, 1987] Al-Khayyal, F. (1987). An implicit enumeration procedure for the general linear complementarity problem. *Mathematical Programming Studies*, 31:1–20.
- [Audet et al., 1997] Audet, C., Hansen, P., Jaumard, B., e Savard, G. (1997). Links between linear bilevel and mixed 0-1 programming problems. *Journal of Optimization Theory and Applications*, 93(2):273–300.
- [Bard e Moore, 1990] Bard, J. e Moore, J. (1990). A branch and bound algorithm for the bilevel programming problem. *SIAM Journal on Scientific and Statistical Computing*, 11(2):281–292.
- [Berge, 1982] Berge, C. (1982). *The Theory of Graphs and its Applications*. Greenwood Press, New York.
- [Bixby et al., 1992] Bixby, R. E., Ceria, S., McZeal, C. M., e Savelsbergh, M. W. P. (1992). Disponível em <http://miplib.zib.de>.
- [Boisvert et al., 1998] Boisvert, R., Pozo, R., Remington, K., Miller, B., e Lipman, R. (1998). Mathematical and Computational Sciences Division of Information Technology Laboratory of National Institute of Standards and Technology, Disponível em <http://math.nist.gov/MatrixMarket/formats.html>.
- [Bomze, 1998] Bomze, I. (1998). On standard quadratic optimization problems. *Journal of Global Optimization*, 13:369–387.
- [Calamai e Vicente, 1993] Calamai, P. H. e Vicente, L. N. (1993). Generating linear and linear-quadratic bilevel programming problems. *SIAM Journal on Scientific Computing*, 14(4):770–782.
- [Chung, 1989] Chung, S. (1989). NP-completeness of the linear complementarity problems. *Journal of Optimization Theory and Applications*, 60:393–399.
- [Cornuéjols et al., 2003] Cornuéjols, G., Li, Y., e Vandenbussche, D. (2003). K-cuts: A variation of gomory mixed integer cuts from the lp tableau. *INFORMS Journal on Computing*, 15:385–396.

- [Cornuéjols et al., 1991] Cornuéjols, G., Sridharan, R., e Thizy, J. M. (1991). A comparison of heuristics and relaxations for the capacitated plant location problem. *EJOR*, 50:280–297.
- [Cottle et al., 1992] Cottle, R., Pang, J., e Stone, R. (1992). *The Linear Complementarity Problem*. Academic Press, New York.
- [Dirkse et al., 2005] Dirkse, S., Ferris, M., e Meeraus, A. (2005). Mathematical programs with equilibrium constraints: Automatic reformulation and solution via constrained optimization. Em *Frontiers in Applied General Equilibrium Modeling*. T.Kehoe, T. Srinivasan and J. Whalley editores, Cambridge University Press, 67-93.
- [Du e Pardalos, 1999] Du, D.-Z. e Pardalos, P. M. (1999). *Handbook of Combinatorial Optimization*. Kluwer Academic Publishers, Dordrecht.
- [Facchinei et al., 1999] Facchinei, F., Jiang, H., e Qi, L. (1999). A smoothing method for mathematical programs with equilibrium constraints. *Mathematical Programming*, 85:107–134.
- [Frangioni, 1995] Frangioni, A. (1995). On a new class of bilevel programming problems and its use for reformulating mixed integer problems. *European Journal of Operational Research*, 82:615–646.
- [Fukushima et al., 1998] Fukushima, M., Luo, Z., e Pang, J. (1998). A globally convergent sequential quadratic programming algorithm for mathematical programs with linear complementarity constraints. *Computational Optimization and Applications*, 10:5–34.
- [Garey e Johnson, 1983] Garey, M. R. e Johnson, D. S. (1983). Crossing number is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 4:312–316.
- [Gavish, 1983] Gavish, B. (1983). Formulations and algorithms for the capacitated minimal directed tree problem. *Journal of ACM*, 30:118–132.
- [Gill et al., 1981] Gill, P., Murray, W., e Wright, M. (1981). *Practical Optimization*. Academic Press, London.
- [Grötschel, 1999] Grötschel, M. (1999). Characterizations of perfect graphs. *OPTIMA Mathematical Programming Society Newsletter*, 62:1–5.
- [Hager, 1984] Hager, W. W. (1984). Condition estimates. *SIAM Journal on Science and Statistical Computing*, 5(2):311–316.
- [Higham, 1988] Higham, N. J. (1988). FORTRAN codes for estimating the one-norm of a real or complex matrix with applications to condition estimation. *ACM Transactions on Mathematical Software*, 14(4):381–396.

- [Higham e Tisseur, 2000] Higham, N. J. e Tisseur, F. (2000). A block algorithm for matrix 1-norm estimation, with an application to 1-norm pseudospectra. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1185–1201.
- [Horst e Pardalos, 1995] Horst, R. e Pardalos, P. M. (1995). *Handbook of Global Optimization*. Kluwer Academic Publishers, Dordrecht.
- [Horst et al., 2000] Horst, R., Pardalos, P. M., e Thoai, N. (2000). *Introduction to Global Optimization 2nd Edition*. Kluwer Academic Publishers, Dordrecht.
- [Horst e Tuy, 1996] Horst, R. e Tuy, H. (1996). *Global Optimization: Deterministic Approaches. 3rd Edition*. Springer, Berlin.
- [ILOG, 2002] ILOG, C. (2002). *Using the CPLEX 8.0 Callable Library*. ILOG Inc., Incline, Village, Nevada.
- [Júdice e Faustino, 1988] Júdice, J. e Faustino, A. (1988). An experimental investigation of enumerative methods for the linear complementarity problem. *Computers and Operations Research*, 15:417–426.
- [Júdice e Faustino, 1991] Júdice, J. e Faustino, A. (1991). A computational analysis of LCP methods for bilinear and concave quadratic programming. *Computers and Operations Research*, 18:645–654.
- [Júdice e Faustino, 1992] Júdice, J. e Faustino, A. (1992). A SLCP method for bilevel linear programming. *Annals of Operations Research*, 34:89–106.
- [Júdice et al., 2002] Júdice, J., Faustino, A., e Ribeiro, I. (2002). On the solution of NP-hard linear complementarity problems. *Top-Sociedad Española de Estadística e Investigación Operativa*, 10(1):125–145.
- [Júdice et al., 2006a] Júdice, J., Faustino, A., Ribeiro, I., e Neves, A. S. (2006a). On the use of bilevel programming for solving a structural optimization problem with discrete variables. Em *Optimization with Multivalued Mappings: Theory, Applications and Algorithms*. S. Dempe e S. Kalashnikov, Springer, 123-142.
- [Júdice et al., 2005] Júdice, J., Serali, H., Ribeiro, I., e Faustino, A. (2005). A complementarity active-set algorithm for mathematical programming problems with equilibrium constraints. *Aceite para publicação no Journal of Optimization Theory and Applications*.
- [Júdice et al., 2006b] Júdice, J., Serali, H., Ribeiro, I., e Faustino, A. (2006b). A complementarity-based partitioning and disjunctive cut algorithm for mathematical programming problems with equilibrium constraints. *Journal of Global Optimization*, 136(1):89–114.

- [Júdice e Vicente, 1994] Júdice, J. J. e Vicente, L. N. (1994). On the solution and complexity of a generalized linear complementarity problem. *Journal of Global Optimization*, 4:415–424.
- [Johnson, 1990] Johnson, C. (1990). *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, London.
- [Johnson e Trick, 1996] Johnson, D. e Trick, A. (1996). Cliques, coloring and satisfiability: Second dimacs implementation challenge. *American Mathematical Society*, 26 da Série Dimacs.
- [Kirchgässner, 1962] Kirchgässner, K. (1962). Ein verfahren zur maximierung linear funktionen in nichtkonvexen bereichen. *Zeitschrift fur Angewandl Mathematik*, 42:22–24.
- [Kojima et al., 1991] Kojima, M., Megiddo, N., Noma, T., e Yoshise, A. (1991). A unified approach to interior point algorithms for linear complementarity problems. Em *Lecture Notes in Computer Science*. Springer-Verlag, Berlin.
- [Kojima et al., 1989] Kojima, M., Mizuno, S., e Yoshise, A. (1989). A polynomial time algorithm for a class of linear complementarity problems. *Mathematical Programming*, 44:1–26.
- [Konno, 1971] Konno, H. (1971). Bilinear programming: Part II - Applications of bilinear programming. Technical report, Department of Operations Research, Stanford University.
- [Locatelli et al., 2004] Locatelli, M., Bomze, I., e Pelillo, M. (2004). The combinatorics of pivoting for the maximum weight clique. *Operations Research Letters*, 32(6):523–529.
- [Luo et al., 1997] Luo, Z., Pang, J., e Ralph, D. (1997). *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, New York.
- [Massaro et al., 2002] Massaro, A., Pelillo, M., e Bomze, I. (2002). A complementary pivoting approach to the maximum weight clique problem. *SIAM Journal on Optimization*, 12(4):928–948.
- [Moler et al., 2001] Moler, C., Little, J. N., e Bangert, S. (2001). *Matlab User's Guide - The Language of Technical Computing*. The MathWorks, Sherborn, Mass.
- [Motzkin e Straus, 1965] Motzkin, T. e Straus, E. (1965). Maxima for graphs and a new proof of a theorem of Túrán. *Canadian Journal of Mathematics*, 17:533–540.
- [Murty, 1983] Murty, K. (1983). *Linear Programming*. Wiley, New York.
- [Murty, 1988] Murty, K. (1988). *Linear Complementarity, Linear and Nonlinear Programming*. Heldermann Verlag, Berlin.

- [Murty e Júdice, 1996] Murty, K. e Júdice, J. (1996). On the complexity of finding stationary points of nonconvex quadratic programming. *Opsearch*, 33:162–166.
- [Outrata et al., 1998] Outrata, J., Kocvara, M., e Zowe, J. (1998). *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints: Theory, Applications and Numerical Results*. Kluwer Academic Publishers, Boston.
- [Scheel e Scholtes, 2000] Scheel, H. e Scholtes, S. (2000). Mathematical programs with complementarity constraints: Stationarity, optimality and sensitivity. *Mathematics of Operations Research*, 25:1–22.
- [Scholtes, 1999] Scholtes, S. (1999). Active set methods for inverse linear complementarity problems. Technical report, Judge Institute of Management Research.
- [Vicente et al., 1996] Vicente, L. N., Savard, G., e Júdice, J. J. (1996). Discrete linear bilevel programming problem. *Journal of Optimization Theory and Applications*, 89:597–614.